

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-307731

(43)Date of publication of application : 17.11.1998

(51)Int.Cl.

G06F 9/46  
H04N 7/24

(21)Application number : 10-052073

(71)Applicant : MATSUSHITA ELECTRIC IND CO LTD

(22)Date of filing : 04.03.1998

(72)Inventor : TANAKA TAKATOSHI  
MAENOBU KIYOSHI  
YOSHIOKA KOSUKE  
HIRAI MAKOTO  
KIYOHARA TOKUZO

(30)Priority

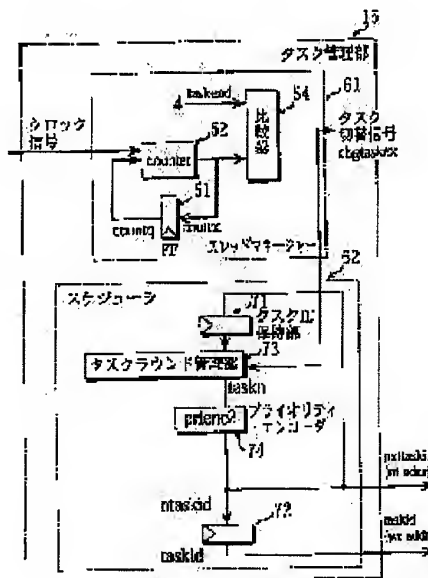
Priority number : 09 49370 Priority date : 04.03.1997 Priority country : JP

(54) PROCESSOR CAPABLE OF EFFICIENTLY EXECUTING ASYNCHRONOUS EVENT TASKS EVEN IN THE CASE EXISTING A LOT OF TASKS TO BE ASYNCHRONOUSLY EXECUTED

(57)Abstract:

**PROBLEM TO BE SOLVED:** To provide a processor which can unequivocally calculate the optimum lower limit value of operating clock number showing how many operating clocks are to be determined for completing the processing of audio out task and video out task.

**SOLUTION:** A counter 52 is a counter setting its initial value to '1' and setting its upper limit value to '4' and counts up count values held in a flip-flop 51 like 1, 2, 3, 4, 1, 2, 3, 4 synchronously with a clock signal. Since this clock signal is used for controlling the execution of instructions by an instruction decoding control part, count-up due to the counter 52 is performed each time this instruction decoding control part once performs instruction execution. A comparator 54 compares the value counted up by the counter 52 with an integral value '4' and when the count value at the counter 52 is coincident with the integer value '4', by turning a task switching signal chg-task-ex into high value, switching is performed so as to complete the next task.



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-307731

(43) 公開日 平成10年(1998)11月17日

(51) Int.Cl.<sup>6</sup>

識別記号

F I

G 0 6 F 9/46

3 4 0

G 0 6 F 9/46

3 4 0 E

H 0 4 N 7/24

H 0 4 N 7/13

Z

審査請求 未請求 請求項の数33 O L (全 46 頁)

(21) 出願番号 特願平10-52073

(22) 出願日 平成10年(1998) 3 月 4 日

(31) 優先権主張番号 特願平9-49370

(32) 優先日 平 9 (1997) 3 月 4 日

(33) 優先権主張国 日本 (J P)

(71) 出願人 000005821

松下電器産業株式会社

大阪府門真市大字門真1006番地

(72) 発明者 田中 卓敏

大阪府門真市大字門真1006番地 松下電器  
産業株式会社内

(72) 発明者 前信 潔

大阪府門真市大字門真1006番地 松下電器  
産業株式会社内

(72) 発明者 吉岡 康介

大阪府門真市大字門真1006番地 松下電器  
産業株式会社内

(74) 代理人 弁理士 中島 司朗

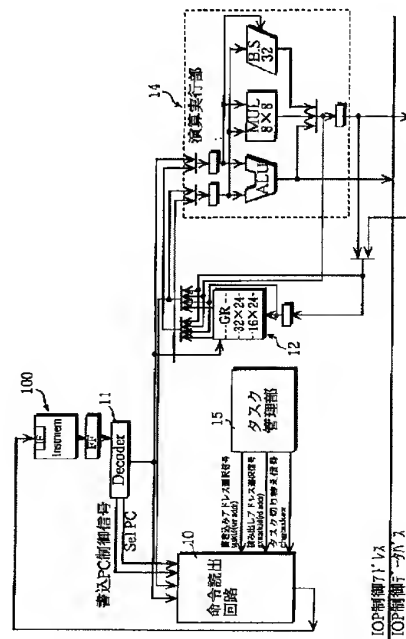
最終頁に続く

(54) 【発明の名称】 非同期に実行すべきタスクが多数あっても、非同期イベントタスクを効率的に実行することができるプロセッサ

(57) 【要約】

【課題】 オーディオアウトタスク、ビデオアウトタスクの処理を完遂させるには動作クロックをいくらに定めればよいかを示す動作クロック数の最適な下限値が一義的に計算できるプロセッサを提供する。

【解決手段】 カウンタ52は、初期値が『1』に設定され、上限値が『4』に設定されたカウンタであり、クロック信号に同期してフリップフロップ51に保持されているカウント値を1,2,3,4,1,2,3,4というようにカウントアップする。このクロック信号は命令解釈制御部11が命令の実行制御のために用いているものであるから、カウンタ52によるカウントアップは、命令解釈制御部11が命令実行を一回行う度になされる。比較器54は、カウンタ52がカウントアップした値と、整数値『4』との比較を行い、もしカウンタ52のカウント値と整数値『4』とが一致すれば、タスク切替信号chg\_task\_exをHigh値にして次のタスクを消化するよう切り換えを行う。



## 【特許請求の範囲】

【請求項1】 n個のタスクを実行対象タスクとしたプロセッサであって、  
n個のタスク識別子を出力し、そのタスクに割り当てられた数の命令が実行されると次のタスク識別子を所定の順序で出力する実行タスク指示手段と、  
出力されるタスク識別子で特定されるタスク中の実行すべき命令を順次指定する命令指定手段と、  
指定された命令を実行する実行手段とを備えることを特徴とするプロセッサ。

【請求項2】 前記実行タスク指示手段は発行された命令をカウンとし、カウント値が実行中タスクに割り当てられた命令数になるとタスク切替信号を発生するタスク切替信号発生器と、  
タスク切替信号が発される度に次順位のタスク識別子を生成して出力するタスク識別子出力部とを備えることを特徴とする請求項1記載のプロセッサ。

【請求項3】 前記命令指定手段は、  
n個のタスクと一対一に対応し、対応するタスクに関して次に読み出すべき先頭アドレスを保持しているn個とのアドレスレジスタと、  
実行タスク指示手段からタスク識別子が出力されると、その識別子と対応するアドレスレジスタを選択するレジスタ選択部と、  
レジスタ選択部で選択されたアドレスレジスタが格納する先頭アドレスを、最初の読出先アドレスとし、タスク識別子が切り替えられるまでの間前記読出先アドレスを順次更新する読出先アドレス生成部と、  
次順のタスクに切り換わる際に、読出アドレス生成部で更新された読出アドレスで、切り換え直前までレジスタ選択部で選択されていたアドレスレジスタの格納する先頭アドレスを書き換えるアドレスレジスタ更新部とを備え、  
前記実行手段は、前記読出先アドレス生成部が生成したアドレスで特定される命令を実行することを特徴とする請求項2記載のプロセッサ。

【請求項4】 前記タスク切替信号発生器は、  
クロックパルスをカウントするカウンタと、  
カウンタのカウント値を現在実行中のタスクに割り当てられた命令数と比較し、一致するとタスク切替信号を出力すると共に前記カウンタのリセットを行う比較器とを備えることを特徴とする請求項2記載のプロセッサ。

【請求項5】 前記タスク切替信号発生器は更に、  
2種以上の命令数を保持する命令数保持部と、  
実行タスク指示手段が指示するタスク識別子によって命令数保持部が保持する一の命令数を選択する命令数選択部とを備え、  
選択された命令数が比較器に比較入力として加えられることを特徴とする請求項4記載のプロセッサ。

【請求項6】 前記命令数保持部が保持する命令数に

は、

第1定数と、第1定数より少ない第2定数があり、  
前記命令指定手段は更にn個のタスクにおいて第2定数を割り当てるべきタスクであるペナルティタスクの識別子を保持するペナルティタスク保持部を備え、  
前記命令数選択部は実行タスク指示手段がタスク識別子を出力すると、当該タスク識別子がペナルティタスクの識別子であるかを判定する判定部と、  
タスク識別子がペナルティタスクを示すと判定されれば、命令数保持部が保持する第2定数を比較器に出力する定数出力部とを備えることを特徴とする請求項5記載のプロセッサ。

【請求項7】 前記タスクには、事象成立の真偽条件の指定を含む事象監視命令が含まれており、  
前記実行手段は読出先アドレス生成部が生成したアドレスにて特定される命令を解読する解読手段と、  
解読手段による解読結果が事象監視命令であれば、事象監視命令において指定された事象成立の真偽を確認する旨の演算を実行する演算手段とを備え、  
前記ペナルティタスク保持部は、  
演算手段による演算結果が事象不成立を示すものならば、事象監視命令を含んでいたタスクのタスク識別子をペナルティタスクの識別子として保持することを特徴とする請求項6記載のプロセッサ。

【請求項8】 前記プロセッサは、その内部にバッファと、メモリとを有する情報処理システムに備えられ、  
前記プロセッサは、内部バッファ及びメモリにおける状態を示す状態レジスタを複数有し、  
前記事象監視命令は、複数の状態レジスタの何れかについての指定と、即値指定とを有し、  
前記演算手段は、  
解読手段による解読結果が事象監視命令であれば、事象監視命令において指定された状態監視レジスタの保持値と、当該命令において指定された即値とを用いた演算を行うことを特徴とする請求項7記載のプロセッサ。

【請求項9】 前記タスク識別子出力部は各タスク識別子を何番目に出力するかを示す出力順位を、n個の全てのタスクについて記憶する順位記憶部と、  
タスク切替信号が発される度に、順位記憶部において出力順位が次順位となるタスク識別子を命令指定手段に出力する選択出力部とを備えることを特徴とする請求項2記載のプロセッサ。

【請求項10】 前記プロセッサにおいて、  
何れかのタスクは緊急扱いを要する旨の緊急宣言命令を含み、  
前記実行手段は命令指定手段が指定した命令を解読する解読手段を備え、  
前記実行タスク指示手段は解読手段による解読結果が緊急宣言命令であれば、当該命令によって指定されるタスクを緊急扱いを要するタスクとしてその識別子を保持す

る緊急タスクレジスタと、

緊急タスクレジスタが緊急タスクの識別子を保持していない状態においてタスク切替信号が発されれば、選択出力部が出力したタスク識別子を実行手段にスルー出力し、

緊急タスクレジスタが緊急タスクの識別子を保持した状態においてタスク切替信号が発されれば、タスク切替信号の $m$ 回の発信につき( $m \geq 2$ )一回の比率にて、緊急タスクレジスタが保持しているタスク識別子を実行手段に出力する出力比率制御部とを備えることを特徴とする請求項9記載のプロセッサ。

【請求項11】 前記プロセッサにおいて、何れかのタスクは緊急扱いを要する旨の緊急宣言命令を含み、

前記実行タスク指示手段は更にプロセッサ外部から入力される所定信号を監視することにより、当該プロセッサ外部が緊急期間内であるか、緊急期間外であるかを判定する監視手段と、

所定信号が緊急期間内である場合に、緊急扱いすべきタスクである緊急タスクの識別子を保持する緊急タスクレジスタと、

緊急期間外においてタスク切替信号が発されれば、選択出力部が未出力のタスクから選んで出力したタスク識別子を実行手段にスルー出力し、

緊急期間内においてタスク切替信号が発されれば、タスク切替信号の $m$ 回の発信につき( $m \geq 2$ )一回の比率にて、緊急タスクレジスタが保持しているタスク識別子を実行手段に出力する出力比率制御部とを備えることを特徴とする請求項9記載のプロセッサ。

【請求項12】 前記順位記憶部は $n$ ビット列を保持する $n$ ビットレジスタを備え、

$n$ ビット列の各ビットは、それぞれがタスク識別子に対応づけられており、

各タスク識別子の出力順位は、それぞれのタスク識別子に対応づけられたタスクビットが当該 $n$ ビット列において最上位或は最下位から何ビット目に位置するかにより表現されることを特徴とする請求項11記載のプロセッサ。

【請求項13】 前記 $n$ ビット列において、タスク識別子が出力済みのタスクに割り当てられたビットは『1』に設定され、未出力タスクに割り当てられたビットは『0』に設定されており、

前記タスク識別子出力部は選択出力部により次順位のタスク識別子が出力されると、 $n$ ビットレジスタが保持する $n$ ビット列のうち、出力されたビットを『1』に設定するビットセット部と、

$n$ ビット列の全ビットがビットセット部により『1』に設定されると、 $n$ ビット列をオールゼロにリセットするビットリセット部とを備え、

前記選択出力部は $n$ ビットレジスタが保持する $n$ ビット列

において『1』から『0』への反転が現れているビット位置に基づいて、次順位に出力すべきタスク識別子を生成すると共に、そのタスク識別子を実行タスク指示手段に出力するプライオリティエンコーダを備えることを特徴とする請求項12記載のプロセッサ。

【請求項14】 前記タスク識別子出力部は緊急タスクレジスタが緊急宣言命令を含むタスクのタスク識別子を保持すれば、 $n$ ビットレジスタが保持している $n$ ビット列において当該緊急タスクに割り当てられたビットを

『1』に設定するマスク部を備えることを特徴とする請求項13記載のプロセッサ。

【請求項15】 前記プロセッサは、動画ストリームを復号するAVデコードシステムに備えられ、

緊急タスクは、

AVデコードシステムにおける動画ストリームの復号が終了と、復号結果を外部に接続されたディスプレイに出力するタスクであり、

監視手段が監視する所定信号とは、映像信号を表示するためのディスプレイにおける水平帰線期間信号、垂直帰線期間信号であることを特徴とする請求項14記載のプロセッサ。

【請求項16】 前記タスク切替信号発生器はクロックパルスをカウントするカウンタと、カウンタのカウント値を現在実行中のタスクに割り当てられた命令数と比較し、一致するとタスク切替信号を出力すると共に前記カウンタのリセットを行う比較器とを備えることを特徴とする請求項11記載のプロセッサ。

【請求項17】 前記タスク切替信号発生器は、2種以上の命令数を保持する命令数保持部と、実行タスク指示手段が指示するタスク識別子によって命令数保持部が保持する一の命令数を選択する命令数選択部とを備え、

選択された命令数が比較器に比較入力として加えられることを特徴とする請求項16記載のプロセッサ。

【請求項18】 前記命令数保持部が保持する命令数には、

第1定数と、第1定数より多い第2定数があり、

前記タスク識別子出力部は次にタスク識別子が出力された際、第2定数を割り当てべき救済対象タスク識別子を保持する救済対象タスク保持部を備え、

前記命令数選択部は監視手段が、プロセッサ外部から入力される所定信号が緊急期間内であると判定した場合、実行タスク指示手段がタスク識別子を出力すると、当該タスク識別子が救済対象タスクを示すものかを判定する判定部と、

タスク識別子が救済対象タスクを示すと判定されれば、命令数保持部が保持する第2定数を比較器に出力する定数出力部とを備えることを特徴とする請求項17記載のプロセッサ。

【請求項19】 前記プロセッサにおいて何れかのタス

クは何れかのタスクの実行を休眠状態に設定すべき旨の自発休眠命令を含み、  
前記実行手段は更に命令指定手段が次に実行すべき命令として指定した命令を解読する解読手段を備え、  
前記実行タスク指示手段は更に解読手段による解読結果が自発休眠命令であれば、当該命令により指示されるタスクを休眠扱いを要するタスクとしてそのタスク識別子を保持する休眠タスクレジスタを備え、  
前記選択出力部は、  
順位記憶部が記憶における次順位のタスク識別子が、休眠タスクのタスク識別子なら、その次順位のタスク識別子を出力することを特徴とする請求項9記載のプロセッサ。

【請求項20】 何れかのタスクは、他のタスクの休眠状態を解除すべき旨の強制解除命令を含み、  
前記実行タスク指示手段は解読手段による解読結果が強制解除命令であれば、休眠タスクレジスタが保持しているタスク識別子を削除する削除部を備えることを特徴とする請求項19記載のプロセッサ。

【請求項21】 前記順位記憶部は、  
nビット列を保持するnビットレジスタと、  
nビット列の各ビットは、それぞれがタスク識別子に対応づけられており、  
各タスク識別子の出力順位は、それぞれのタスク識別子に対応づけられたタスクビットが当該nビット列において最上位或は最下位から何ビット目に位置するかにより表現されることを特徴とする請求項20記載のプロセッサ。

【請求項22】 前記nビット列において、  
タスク識別子が出力済みのタスクに割り当てられたビットは『1』に設定され、未出力タスクに割り当てられたビットは『0』に設定されており、  
前記実行タスク出力部は選択出力部により次順位のタスク識別子が出力されると、nビットレジスタが保持するnビット列のうち、出力されたビットを『1』に設定するビットセット部と、  
nビット列の全ビットがビットセット部により『1』に設定されると、nビット列をオールゼロにリセットするビットリセット部とを備え、  
前記選択出力部はnビットレジスタが保持するnビット列において『1』から『0』への反転が現れているビット位置に基づいて、次順位に出力すべきタスク識別子を生成すると共に、そのタスク識別子を実行タスク指示手段に出力するプライオリティエンコーダを備えることを特徴とする請求項21記載のプロセッサ。

【請求項23】 前記実行タスク指示手段は緊急タスクレジスタが自発休眠命令を含むタスクのタスク識別子を保持すれば、nビット列において当該休眠状態タスクに割り当てられたビットを『1』にマスクするマスク部と、

解読手段による解読結果が強制解除命令であれば、nビット列において当該命令にて指定されるタスクのタスク識別子に割り当てられたビットを『0』にしてマスクを解除するアンマスク部とを備えることを特徴とする請求項22記載のプロセッサ。

【請求項24】 各タスクに割り当てられたサイクル数を保持するサイクル数保持手段と、  
クロックをカウントするカウンタと、  
カウンタのカウント値がサイクル数保持手段の最大サイクル数に一致すると、タスク切り替え信号とカウンタリセット信号を発する比較手段と、  
予め決められたタスク選択の順序を有し、タスク切替信号が入力されると、タスクの選択を切り替える選択手段と、  
選択手段が一のタスクを選択している間、そのタスクの命令を順次指定する命令指定手段と、  
選択手段が選択したタスクの、命令指定手段が指定する命令を実行する実行手段とを備え、  
前記サイクル数は一のタスクが実行され始めてから次のタスクに切り替えられるまでの最大サイクル数であり、  
全タスクの最大サイクル数の合計時間は非同期事象の発生周期よりも短く設定されていることを特徴とするプロセッサ。

【請求項25】 実行対象タスクの一つは前記非同期事象に関する処理が割り当ててあり、そのタスクに割り当てられたサイクル数と非同期事象発生周期内にそのタスクが繰り返し選択される回数との積で与えられるサイクル数が非同期事象に対して所要処理量を上回っていることを特徴とする請求項24記載のプロセッサ。

【請求項26】 前記プロセッサは、動画ストリーム及びオーディオストリームを復号するデコード部と、デコード部が復号処理に用いるローカルバッファと、外部メモリとを有するAVデコードシステムに備えられ、  
前記タスク数nは4であり、  
n個のタスクにおける第1タスクは、AVデコードシステム外部から入力されてくるビットストリームから動画ストリーム及びオーディオストリームを抽出して、抽出された動画ストリーム及びオーディオストリームを外部メモリに書き込むタスクであり、  
第2タスクは、AVデコードシステムにおけるデコード部の進捗に応じて、次に処理すべき動画ストリーム及びオーディオストリームを外部メモリからデコード部に供給するタスクであって、デコード結果を外部メモリに格納するタスクであり、  
第3タスクは、  
デコード部のデコード結果のうち動画データを外部メモリから読み出して、外部に接続されたディスプレイの同期信号に同期させつつ出力するタスクであり、第4タスクは、  
デコード部のデコード結果のうち音声データを外部メモ

リから読み出して、外部に接続されたスピーカ装置に所定の周期で出力するタスクであり、

前記サイクル数保持手段は、

第1～第4タスクに割り当てられたサイクル数として4サイクルを保持することを特徴とする請求項2記載のプロセッサ。

【請求項27】  $n$ 個のタスクを実行対象タスクとしたプロセッサであって決められたサイクル数でタスクを順次選択するタスク選択手段と、

タスクと一対一の関係で $n$ 個の命令指定情報を有し、タスク選択手段によってタスクが選択されるとそれに対応した命令指定情報を有効にすると共に、その情報を起点として、次に読み出すべき命令を指定する情報を動的に生成する命令指定手段と、

命令指定手段にて指定された命令を読み出し実行する実行手段とを備えることを特徴とするプロセッサ。

【請求項28】 前記命令指定手段は、

$n$ 個のタスクと一対一に対応し、対応するタスクに関して次に読み出すべきアドレス値を命令指定情報として保持している $n$ 個とのアドレスレジスタと、

タスク選択手段が選択したタスクに対応するアドレスレジスタを選択し、そのアドレス値を出力させるレジスタ選択部と、

レジスタ選択部によりアドレスレジスタが選択されると、当該アドレスレジスタが保持しているアドレス値をカウント初期値として保持するカウント値レジスタと、カウント値レジスタが保持しているカウント値を、サイクル毎に、インクリメントするインクリメントと、インクリメントされたカウント値を前記次に読み出すべき命令を指定する情報の更新値として保持する読出先アドレス保持部とを備えることを特徴とする請求項27記載のプロセッサ。

【請求項29】 前記命令指定手段はインクリメントがインクリメントしたカウント値をスルー出力して読出先アドレス保持部に保持させると共に、タスク選択手段がタスクを選択すると、当該タスクに対応するアドレスレジスタのアドレス値を選択出力して読出先アドレス保持部に保持させる第1セレクタと、

次順のタスクに切り換わる際に、カウント値レジスタが保持しているカウント値を用いて、切り換え直前までレジスタ選択部で選択されていたアドレスレジスタが格納するアドレス値を書き換える第1書換部とを備えることを特徴とする請求項28記載のプロセッサ。

【請求項30】 前記タスク切り換えを指定する旨の自発切換命令が含まれており、

前記実行手段は読出先アドレス保持部が保持しているアドレスにて特定される命令を解読する解読手段を備え、命令指定手段は、

解読手段による解読結果が自発切換命令であれば、読出先アドレス保持部が保持している命令指定情報を用い

て、切り換え直前まで選択部で選択されていたアドレスレジスタの格納するアドレス値を書き換える第2書換部を備え、

前記第1セレクタは、

解読手段による解読結果が自発切換命令であれば、当該タスクに対応するアドレスレジスタのアドレス値を選択出力して読出先アドレス保持部に保持させることを特徴とする請求項29記載のプロセッサ。

【請求項31】 前記タスクには絶対アドレスにて分岐先を指定した分岐命令が含まれており、

前記命令指定手段は解読手段による解読結果が分岐命令であれば、当該分岐命令に含まれている絶対アドレスを用いて、切り換え直前まで選択部で選択されていたアドレスレジスタの格納するアドレス値を書き換える第3書換部を備えることを特徴とする請求項30記載のプロセッサ。

【請求項32】 前記タスクには間接アドレス指定にて分岐先を指定した分岐命令が含まれており、

前記命令指定手段は解読手段による解読結果が間接アドレス指定型の分岐命令であり、実行手段における命令実行によって分岐先アドレスが算出されれば、当該分岐先アドレスを用いて切り換え直前まで選択部で選択されていたアドレスレジスタの格納するアドレス値を書き換える第4書換部を備えることを特徴とする請求項31記載のプロセッサ。

【請求項33】 前記実行手段は $n \times m$ 個の汎用レジスタから構成され、各タスクに $m$ 個の汎用レジスタを割り当てている汎用レジスタセットと、

タスク選択手段によりタスクが選択されると、それに対応する $m$ 個の汎用レジスタを用いて選択されたタスクに含まれている命令についての演算を実行する演算実行部とを備えることを特徴とする請求項32記載のプロセッサ。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】MPEGストリームを始めとする可変符号長のマルチメディアデータを再生するAVデコーダに備えられ、専らAVデコーダ内部のメインプロセッサの周辺制御を担うプロセッサに関する。

【0002】

【従来の技術】MPEGストリームの再生技術はマルチメディア社会の基盤技術の一つでもあり、近年におけるその需要の高まりには目を見張るものがある。MPEGストリームの再生技術の新たな活躍の舞台として注目を浴びているのは、インタラクティブな動画再生、音声再生を可能とする民生機器の分野であり、この分野にて成功を収めるべく、メーカー各社の技術者はMPEGストリームの再生を高度になしえるAVデコーダの研究開発に心血を注いでいる。

【0003】MPEGストリームの再生技術にて行うべき処

理には様々なものがあるが、それらはコア処理と、非同期イベント処理とに大別されるといわれる。コア処理とは、逆量子化、逆離散余弦変換、動き補償等、縦16×横16個の画素からなるマクロブロックを対象とした処理である。毎秒4050個ものマクロブロックを処理せねばならないので(4050=30フレーム×30スライス×45マクロブロック)、その演算量は膨大なものとなる。コア処理の実行には、パイプライン処理が好適とされるが、ハードウェア規模の増大を厭わない場合は、解読器や演算器を複数にしてコア処理の負荷を分散させるのが良い。

【0004】非同期イベント処理とは、複数の要因が重なることにより特定の事象が成立している期間において集中的に行うべき処理、或は、所定の周期をもって間欠的に行うべき処理であり、コア処理と同期して行なえない処理の総称をいう。AVデコーダにおいて、このような非同期イベント処理に相当する処理には、記録媒体や通信媒体からのMPEGストリーム入力に関するもの(1)、AVデコーダから映像再生機器、音声再生機器への出力に関するもの(2)、AVデコーダと、AVデコーダ外部に接続された拡張メモリとの入出力に関するもの(3)の三種類がある。

【0005】記録媒体からのストリーム入力に関する非同期イベント処理(1)は更に、光ディスク等の記録媒体や通信媒体からMPEGストリームを取り込んで、エレメンタリストリームを抽出する抽出処理(1-1)、抽出されたエレメンタリストリームを拡張メモリとして接続されているSDRAMに書き込む書込処理(1-2)がある。再生出力に関する非同期イベント処理(2)は更に、復号が済んだ動画ストリーム、オーディオストリームを映像信号、音声信号にそれぞれ変換してディスプレイ、スピーカに出力する処理(2-1)と、MPEGストリームのプライベートストリームとして出力されてくる副映像を描画し、これを動画データと混合して映像信号に字幕を合成する処理(2-2)を含む。

【0006】拡張メモリとの入出力に関する非同期イベント処理(3)は更に、逆量子化、逆離散余弦変換、動き補償等の処理が施されたデータが内部バッファに蓄積されるのを監視し、その蓄積量が一定量になった段階でSDRAMにまとめて書き込む処理(3-1)、逆量子化、逆離散余弦変換、動き補償の進捗に応じて、次に処理すべきデータを外部に接続されたSDRAMから内部バッファに間欠的に補充する補充処理(3-2)がある。

【0007】以上の処理の他にも、操作者からの操作に応じて行うべき処理はこの非同期イベント処理に分類される。単にMPEGストリームの再生だけではなく、操作者の対話性を重視した再生装置の応用システムを設計する場合やホストコンピュータとの同期制御を前提にしたシステムを設計する場合は、更に多くの非同期イベント処理をAVデコーダに処理させる必要がある。

【0008】上記の処理のうち、処理(2-1)のオーディ

オに関する処理は、オーディオアウトタスクと呼ばれ、90 $\mu$ secという間欠的な周期を有する。また処理(2-1)のうち、ビデオに関する処理はビデオアウトタスクと呼ばれ、ディスプレイの水平同期信号の間欠的な周期に基づいた50 $\mu$ secにおいて、一ライン分の画像の処理を完遂させねばならない。

【0009】このようにビデオアウトタスク、オーディオアウトタスクに実行周期が与えられるのは、これらのタスクの処理が完遂されないと、動画・音声時間が時間の流れに沿って円滑に再生されないからである。即ち、ビデオアウトタスク、オーディオアウトタスクの処理を50 $\mu$ sec、90 $\mu$ secという周期内に完遂することが、動画再生、音声再生のリアルタイム性を満たすための最大要件となる。

【0010】従来のMPEGストリーム再生技術においては、以上の非同期イベント処理を汎用プロセッサに行わせる場合、非同期事象の発生や所定周期の到来を割込信号により汎用プロセッサに通知する。通知後、汎用プロセッサは分岐命令を用いて非同期イベント処理へと分岐する。

【0011】

【発明が解決しようとする課題】しかしながら割込信号を用いて汎用プロセッサに非同期イベント処理を実行させるという従来の方式では、オーディオアウトタスク、ビデオアウトタスクの処理を完遂させるには動作クロックをいくらに定めればよいかを示す動作クロック数の最適な下限値が一義的に計算できないという問題点がある。

【0012】最適な下限値が計算できないため、大まかな見積もりで動作クロック数を決定せざるを得ず、動作クロック数をどうしても高めに設定してしまう傾向がある。従来の方式では、どのような計算法で動作クロックを計算していたかを以下に説明する。周期が異なる非同期イベントタスクとして90 $\mu$ secの実行周期を有するオーディオアウトタスクと、50 $\mu$ secの実行周期を有するビデオアウトタスクを実行せねばならないものとする。この場合、オーディオアウトタスクは0 $\mu$ sec、90 $\mu$ sec、180 $\mu$ sec、270 $\mu$ sec、360 $\mu$ sec、450 $\mu$ secというように90 $\mu$ sec毎にオーディオストリームの復号を完遂せねばならず、ビデオアウトタスクは0 $\mu$ sec、50 $\mu$ sec、100 $\mu$ sec、150 $\mu$ sec、200 $\mu$ sec、250 $\mu$ sec、300 $\mu$ sec、350 $\mu$ secというように50 $\mu$ sec毎に動画ストリームの復号を完遂せねばならない。従来の汎用プロセッサは、2つの実行周期の到来を割込信号にて通知して、それからビデオアウトタスク、オーディオアウトタスクを処理することになる。

【0013】割込信号によって実行時期の到来を知ってから、オーディオアウトタスク、ビデオアウトタスクを起動しようとする、各タスクが処理を完遂せねばならないタイムリミットは、自身を起動するための割込信号

の発生タイミングから、次のタスクを起動するための割込信号の発生タイミングまでに定められる。図57は、これらの周期が割込信号により汎用プロセッサに通知されて、実行する場合を想定したタイミングチャートである。本図において、オーディオアウトタスク・ビデオアウトタスクの処理完遂のためのタイムリミットがどのように現れるかを説明する。

【0014】パルスP1の立ち上がり及びパルスP5の立ち上がりが発生して、ビデオアウトタスク、オーディオアウトタスクを起動する旨が汎用プロセッサに通知されると、次の割込信号であるパルスP2の立ち上がりが通知されるまでの50 $\mu$ sec期間がビデオアウトタスク及びオーディオアウトタスクの処理を完遂させるためのタイムリミットとなる。

【0015】一方、パルスP6の立ち上がりが発生して、オーディオアウトタスクを起動する旨が汎用プロセッサに通知されると、次の割込信号であるパルスP3の立ち上がりが通知されるまでの僅か10 $\mu$ secの期間がオーディオアウトタスク完遂のためのタイムリミットとなる。タイムリミットから動作クロックを計算する方式では、10 $\mu$ secのような極めて短いタイムリミット、即ち、タイムリミットのワーストケースにおいてもオーディオアウトタスク、ビデオアウトタスクを完遂するだけの動作クロック数が要求される。そうすると動作クロックを一段と高いものに決定せざるを得ない。しかし安易に動作クロックを高速にすれば、消費電力の増大も著しくなり、民生機器の用途に適さなくなる。

【0016】本発明の目的は、所定の周期で完遂せねばならない処理量が予め定められている非同期イベント処理が複数ある場合に、それぞれの非同期イベントタスクにどれだけのサイクル数を割り当てればよいかを示す最適な下限値を一義的に計算することできると共に、それらの最適な下限値から全タスクを動作させるための動作サイクル数の下限値を一義的に計算することができるプロセッサを提供することである。

【0017】

【課題を解決するための手段】上記目的は、n個のタスクを実行対象タスクとしたプロセッサであって、n個のタスク識別子を出力し、そのタスクに割り当てられた数の命令が実行されると次のタスク識別子を所定の順序で出力する実行タスク指示手段と、出力されるタスク識別子で特定されるタスク中の実行すべき命令を順次指定する命令指定手段と、指定された命令を実行する実行手段とを備えるプロセッサにより達成される。

【0018】

【発明の実施の形態】

（第1実施形態）I/Oプロセッサについての説明を開始する前に、AVデコーダがどのような内部構成を有しており、その構成において本発明に係るI/Oプロセッサがどのような役割を果たしているかを説明する。図1にAVデ

コーダの内部構成を示す。

【0019】AVデコーダは、コア処理を行うデコーダコア部と、複数の非同期イベントタスクを行うI/O部ととなる。デコーダコア部は、setup部104、VLD部105、IQ/IDCT部106、動き補償部107を含み、I/O部は、ストリーム入力部101、バッファメモリ102、ビットストリームFIFO103、ビデオ出力部108、オーディオ出力部109、I/Oプロセッサ113、バッファメモリコントローラ110、RAMコントローラ111、FIFOコントローラ112、ホストI/O部115を備えている。またバスとしてバッファメモリバス121、SDRAMバス122、ビットストリームバス123、IOPコントロールバス124が設けられており、拡張メモリとしてSDRAM300と接続している。

【0020】(1.)MPEGストリームの復号処理の概要

次に図2を参照して、MPEGストリームの復号処理の概要について簡単に説明しておく。本実施形態においてMPEGストリームは、動画ストリーム、オーディオストリーム、副映像ストリームからなる。このうち動画ストリームは、圧縮された動画画像データであり、復号処理において、差分算出の基準となる画像（以下「参照画像」と呼ぶ）と、算出された差分を符号化した画像（以下「差分画像」と呼ぶ）とを加算することにより、動画画像に復元される。

【0021】図2において、MPEGストリーム内の動画ストリームのデータ構造は、階層的に表現されている。第1層はMPEGストリームの階層であり、第2層は1秒長の動画の階層である。第3層は1フレームの階層であり、第4層は1スライスの階層である。第5層はマクロブロックの階層である。破線C1に示す第1層と第2層との対応関係を参照すると、MPEG(Motion Picture Image Coding Experts Group)において1秒長の動画は、NTSC方式における30フレームの画像（PAL方式では25フレームの画像）にて構成されることがわかる。また、各フレームの画像は、Iピクチャ（図中のI0）、Pピクチャ（図中のP3）、Bピクチャ（図中のB1,B2）という3つのタイプがあることもわかる。

【0022】ここでIピクチャとは、圧縮された画像データであるが、一フレーム分の輝度成分及び色差成分を含むものをいう。Pピクチャ（Predictive-Picture）又はBピクチャ（Bidirectionally predictive Picture）は差分画像と呼ばれるものである。ここでPピクチャとは、過去方向に位置するフレームとの差分からなる差分画像であり、Bピクチャとは、過去方向及び未来方向に位置するフレームとの差分からなる差分画像である。

【0023】Bピクチャ及びPピクチャがどのような単位で作成されるかであるが、破線C2に示す第2層と第3層との対応関係を参照すると、NTSC方式においてこれらの画像は30本のスライスというデータにて構成され、PAL方式では36本のスライスというデータにて構成されるこ



とがわかる。破線C3に示す第3層と第4層との対応関係を参照すると、各スライスは、45のマクロブロックにて構成されることがわかる。マクロブロックとは、横16画素×縦16画素の輝度成分、色差成分にて構成される。破線C4に示す第4層と第5層との対応関係を参照すると、横16×縦16の輝度成分からなる輝度ブロックと、横8×縦8の青色色差成分からなる青色差ブロック(Cbブロック)と、横8×縦8の赤色色差成分からなる赤色差ブロック(Crブロック)とを含むことがわかる。本マクロブロックは、AVデコーダにおける画像復号の一単位となる。

【0024】マクロブロックより下は、AVデコーダの構成要素間がマクロブロックに含まれている輝度ブロック、色差ブロックをどのように処理するかを示すタイミングチャートになっている。マクロブロックに対してどのような復号化がなされるかであるが、マクロブロックに対する復号化は、圧縮動画データを変長復号(Variable Length codeDecoding;以降VLDと略す)して、6つの空間周波数成分データ、ヘッダ情報、動きベクトルを得る。

【0025】その後、6つの空間周波数成分データに対して逆離散余弦変換処理(Discrete Cosine Transform: DCT)を行って低周波数帯に位置する空間周波数成分と、高周波数帯に位置する空間周波数成分とに分離し、高周波数帯を切り捨てて、低周波数帯に位置する空間周波数成分に対して逆量子化処理を行うことによりなされる。符号化されたマクロブロックは、逆量子化及び逆離散余弦変換処理がなされた後、動きベクトルに基づいた動き補償により、映像として表示される。動きベクトルとは、前後のフレームの画像と比較して相関性の最も高い箇所を指し示す情報をいう。つまり動きベクトルは画像内の人物像や建造物像がフレームの前後でどう動いたかをブロック単位で表現している。

【0026】動き補償とは、差分の前後に表示されるべき参照画像と、当該差分とを足し合わせて(ブレンドして)一枚の完結した表示用画像を得ることをいう。以上のMPEG準拠の動画データの符号化・復号化技術のうち、本明細書で特に引用する内容は以上の通りである。より詳細な技術内容については、株式会社アスキー「ポイント図解式最新MPEG教科書」などの公知文献を参照されたい。

【0027】図2におけるタイミングチャートは、デコーダコア部におけるsetup部104、VLD部105、IQ/IDCT部106、動き補償部107の処理がどのようなタイミングが行われるか説明する際に引用するものとする。以上でMPEGストリームの復号処理の概要について説明を終え、続いてAVデコーダの構成要素についての説明を開始する。

【0028】(2.)AVデコーダの構成要素

ストリーム入力部101は、記録媒体や通信媒体からMPEGストリームが取り出されると、MPEGストリームをAVデ

コーダ内部に取り込み、1/0プロセッサ113による制御に応じてバッファメモリバス121に出力する。バッファメモリ102は、バッファメモリコントローラ110の制御に従って、ストリーム入力部101が取り込んだMPEGストリームを保持する。またMPEGストリームを出力するようバッファメモリコントローラ110に命じられると、これまでに格納しているMPEGストリームをSDRAMバス122に出力する。

【0029】ビットストリームFIFO103は、MPEGストリームに含まれているエレメンタリストリームがSDRAMバス122に出力されると、FIFOコントローラ112による制御に従って出力されたエレメンタリストリームを取り込む。ビットストリームFIFO103は取り込んだエレメンタリストリームを先入れ先出し方式に保持する。このようにして保持したエレメンタリストリームをFIFOコントローラ112の制御に従って、ビットストリームバス123に出力する。

【0030】setup部104は、ビットストリームFIFO103に保持されているMPEGストリームからエレメンタリストリームが抽出されるのを待ち、そのエレメンタリストリームが動画ストリーム或はオーディオストリームなら、VLD部105による復号によりそのヘッダ部が伸長されるのを待つ。ヘッダが伸長されるとこれの解析処理e1を図2に示すように行う。またエレメンタリストリームが動画ストリームなら動きベクトルの抽出e2を行う。その後、逆量子化、逆離散余弦変換、動き補償等が行われている間、音声ストリームの復号処理e3を行う。

【0031】VLD部105は、ビットストリームFIFO103に格納されているエレメンタリストリームが動画ストリームであり、これを構成するマクロブロックがビットストリームバス123に出力されると、マクロブロックに含まれている4つの輝度ブロックY0,Y1,Y2,Y3と、2つの色差ブロックCb,Crとに対して可変符号長デコードt21,t22,t23,t24,t25,t26を行う。

【0032】IQ/IDCT部106は、可変符号長デコードがなされた4つの輝度ブロックと、2つの色差ブロックとに対して逆量子化及び逆離散余弦変換を行う。動き補償部107は、IQ/IDCT部106により逆量子化及び逆離散余弦変換がなされると、これらの処理がなされた輝度ブロック及び色差ブロックに対応する参照画像(Y0,Y1),(Y2,Y3),(Cb,Cr)をAVデコーダ外部に接続されたSDRAM300から読み出し、輝度ブロック及び色差ブロックと、参照画像とをブレンドする。そしてそのブレンド結果に対してハーフペル補間を行うことにより動き補償を行う。その後、RAMコントローラ111を制御することにより動き補償の結果をSDRAM300に書き込ませる。

【0033】ビデオ出力部108は、動き補償部107により参照画像とのブレンドとハーフペル補間とがなされた一フレームの画像を映像信号に変換して外部に接続

されたテレビ受像器等のディスプレイ装置に出力する。オーディオ出力部109は、setup部104により復号されたオーディオストリームを音声信号に変換して外部に接続されたスピーカ装置に出力する。

【0034】バッファメモリコントローラ（図中のBMコントローラ）110は、I/Oプロセッサ113周辺におけるストリーム入力部101、ビデオ出力部108、オーディオ出力部109間のアクセス調停を行うバッファメモリインターフェイスと、バッファメモリ102と、ストリーム入力部101、ビデオ出力部108、オーディオ出力部109間のDMA転送機能を有するDMAコントローラとからなる。

【0035】RAMコントローラ111は、SDRAM300に対してのバーストリードと、バーストライトとが可能なSDRAMインターフェイスと、SDRAM300-バッファメモリ102間、動き補償部107-バッファメモリ102間のDMA転送機能を有するDMAコントローラとからなる。FIFOコントローラ112は、ディアルポートRAMと、当該RAMのRead/Writeを制御するコントローラと、ビットストリームFIFO103におけるアクセスアドレスを示すポインタを管理するポインタ管理機能とを有する。

【0036】I/Oプロセッサ113は、AVデコーダにおける6本もの非同期イベントタスクを6本のスレッドに割り当てることにより、非同期イベントタスクを時分割多重にて実行する。各非同期イベントタスクに割り当てたスレッドは、I/Oプロセッサ内部の全ての構成要素が同期制御のために用いているクロック信号のサイクル数にて表現され、その長さは原則4サイクルであるものとする。

【0037】(3.) I/Oプロセッサに実行が課された非同期イベントタスクとは、どのような処理内容であるかAVデコーダにおいてI/Oプロセッサに実行が課された非同期イベントタスクが、どのような処理内容であるかを逐一説明する。

・ホストI/Oタスク

本タスクは、ホストコンピュータとの通信と、操作者との対話に応じた二次元グラフィックスの描画に関するタスクである。具体的には、ホストI/O部115を介したホストコンピュータとの通信処理と、操作者による操作に応じて二次元グラフィックスを描画し、ビデオ出力部108に出力する処理とからなる。

・パージングタスク

本タスクはバッファメモリ102に入力されたMPEGストリームのパージング処理と、MPEGストリームから動画ストリーム、オーディオストリーム、副映像ストリームといったエレメンタリストリームの抽出を行う抽出処理に関する。

【0038】パージングタスクの実行時において、I/Oプロセッサ113は外部からストリーム入力部101に

10

20

30

40

50

ではBMバス)121に出力させる。バッファメモリコントローラ110を制御することによりバッファメモリバス121に出力されたMPEGストリームをバッファメモリ102に書き込ませる。その後I/Oプロセッサ113は、バッファメモリコントローラ110を制御することによりMPEGストリームをバッファメモリバス121に出力させ、MPEGストリームからエレメンタリストリームを抽出する。バッファメモリコントローラ110を制御することにより抽出されたエレメンタリストリームをバッファメモリ102に書き込ませる。その結果、バッファメモリ102には、エレメンタリストリームとしてビットストリーム、オーディオストリーム、副映像ストリームが格納された状態となる。

・オーディオストリーム転送制御タスク

オーディオストリーム転送制御タスクはオーディオストリームに関する全ての転送制御からなるタスクである。

【0039】オーディオストリーム転送制御タスクに従って、I/Oプロセッサ113はバッファメモリコントローラ110を制御することによりバッファメモリ102にエレメンタリストリームとして格納されているオーディオストリームをSDRAMバス122に出力させる。その後、RAMコントローラ111を制御することによりSDRAMバス122に出力されたオーディオストリームをSDRAM300に書き込ませる。setup部104がビットストリームFIFO103に格納されているオーディオストリームの復号を開始すると、その復号処理がどれだけ進捗したかを監視し、オーディオストリームの残量が所定値以下となると、RAMコントローラ111を制御して、SDRAM300に格納されているオーディオストリームを読み出させてSDRAMバス122に出力させる。FIFOコントローラ112を制御してこのように出力されたオーディオストリームをビットストリームFIFO103に書き込ませる。

【0040】このようなビットストリームFIFO103への書き込みにより、オーディオストリームの復号の進捗に応じたオーディオストリームの補充を行う。デコードが済むと、I/Oプロセッサ113はFIFOコントローラ112を制御することによりデコード済みオーディオストリームをSDRAMバス122に出力させる。バッファメモリコントローラ110を制御して、SDRAMバス122に出力されたオーディオストリームをバッファメモリ102に書き込ませる。

・動画ストリーム転送制御タスク

動画ストリーム転送制御タスクは動画ストリームに関する全ての転送制御からなるタスクである。

【0041】I/Oプロセッサ113は、バッファメモリコントローラ110を制御することによりバッファメモリ102に格納されている動画ストリームをSDRAMバス122に出力させ、RAMコントローラ111を制御することによりSDRAMバス122に出力された動画ストリームをSDRAM300に書き込ませる。VLD部105、IQ/IDC

T部106、動き補償部107がビットストリームFIFO103に格納されている動画ストリームの復号を開始すると、動画ストリームの復号がどれだけ進捗したかを監視し、復号すべき残量が所定値以下となると、RAMコントローラ111を制御して、SDRAM300に格納されている動画ストリームを読み出させて、FIFOコントローラ112にビットストリームFIFO103に書き込ませる。このようなビットストリームFIFO103の書き込みにより、復号処理の進捗に応じた動画ストリームの補充を行う。

#### ・ビデオアウトタスク

ビデオアウトタスクは映像出力に関する出力制御からなるタスクである。

【0042】本非同期イベントタスクの実行時においてI/Oプロセッサ113はRAMコントローラ111を制御することにより、VLD部105、IQ/IDCT部106、動き補償部107による処理が済んでSDRAM300に格納されている動画データをSDRAMバス122に出力させ、バッファメモリコントローラ110を制御して、SDRAMバス122に出力された動画ストリームをバッファメモリ102に書き込ませる。バッファメモリコントローラ110を制御して、SDRAM300に書き込まれた動画ストリームをSDRAMバス122に出力させ、ビデオ出力部108に映像信号に変換させる。それと共に副映像ストリームを展開して得られた副映像と、二次元グラフィックスとを混合してビデオ出力部108に出力させる。

#### ・setup1/0タスク

setup部104とI/Oプロセッサ113とが通信する際に実行すべき処理からなる。

【0043】当該コマンドがホストコンピュータが備えているレジスタを読み書きするためのコマンドなら、I/Oプロセッサ113を制御して当該読み書きのための処理と、当該コマンドがVLD部105、IQ/IDCT部106の初期化コマンドなら、VLD部105、IQ/IDCT部106をリセットする処理とからなる。以上の非同期イベントタスクは、個々の処理負荷は軽い、間欠的な起動が要求される。特にビデオアウトタスクは水平ブランキング期間の50μsec毎の起動が要求され、オーディオアウトタスクは90μsec毎の起動が要求される。

【0044】(4.) I/Oプロセッサがどのように構成されているか

I/Oプロセッサの第1実施形態を図面を参照しながら説明する。上述したように、I/Oプロセッサはそれぞれの非同期イベントタスクを実行すべき事象が成立したか否かの監視を行っており、事象が成立した非同期イベントタスクをより優先的に実行する。本実施形態において優先実行の手順までも言及すると、説明が煩雑になるのでこの優先実行についての説明は第3実施形態において行うものとする。第1実施形態では、I/Oプロセッサの基本構成、即ち、AVデコーダにおける6本もの非同期イベ

ントタスクを時分割多重にて実行する構成について説明する。

【0045】図3はI/Oプロセッサの内部構成を示す図であり、本図に示すようにI/Oプロセッサは命令メモリ100、命令読出回路10、命令解読制御部11、レジスタセット12、演算実行部14、及びタスク管理部15から構成される。命令メモリ100は、上述した六本の非同期イベントタスクを構成する命令を記憶する。図8は、命令メモリ100がどのように命令列を記憶しているかを示す図である。本図において各命令には、0-0、0-1、0-2といった識別子が付されている。この識別子のうち、上位の数は、当該命令が6本の非同期イベントタスクの何れに属するかを示し、下位の数は、属する非同期イベントタスクにおいて当該命令が何番目に位置するかを示す。以降の説明において、命令メモリ100における各命令の配置アドレスは、この識別子を用いて表すものとする。

【0046】命令メモリ100内に記憶されている命令は、全てのオペレーションが一命令語長に収まるよう命令フォーマットが規定されている。本実施形態において命令語長は16bitに規定されており、その命令フォーマットを図9～図11に示す。図9は、レジスタをオペランドに指定した算術演算命令の命令フォーマットを示す。本命令フォーマットでは、2bit～4bitに第1オペランドに指定すべきレジスタ番号を指定でき、5bit～7bitには、第2オペランドに指定すべきレジスタ番号を指定できる。8bit～10bitには、演算結果の格納先に指定すべきレジスタ番号を指定でき、11bitから15bitまでの5bitにオペレーション内容が指定される。下段の表は、当該オペレーション内容の指定が可能な5bitのうち、上位2bitと、下位3bitの組み合わせと、その組み合わせにて表現されるオペレーション内容とを示している。例えば上位2bitが『00』であり、下位3bitが『000』である場合、その組み合わせは加算演算のオペレーションを表現しており、上位2bitが『01』であり、下位3bitが『000』である場合、その組み合わせは減算演算を表現している。上位2bitが『00』であり、下位3bitが『001』である場合、その組み合わせは左シフト演算（図中の<<は左シフト演算の意味である）を表現しており、上位2bitが『01』であり、下位3bitが『001』である場合、その組み合わせは右シフト演算（図中の>>は右シフト演算の意味である）を表現している。

【0047】図10は、レジスタにて読出先アドレス、書込先アドレスを指定したメモリロード命令／メモリスストア命令の命令フォーマットを示す。この命令フォーマットでは、2bit～4bitに第1オペランドに指定すべきレジスタ番号を指定でき、5bit～7bitに読出元に指定すべきレジスタ番号又は演算結果の格納先に指定すべきレジスタ番号を指定できる。10bitから12bitまでの3bitにオペレーション内容が指定され、13bitから15bitまでの3b

itに3bit長の即値が指定される。

【0048】下段の表は、3bitによるオペレーション内容の指定のうち、上位1bitと、下位2bitの組み合わせと、その組み合わせにて表現されるオペレーション内容とを示している。例えば上位1bitが『1』であり、下位2bitが『10』である場合、読出元オペランドsrc\_REGで指定されたレジスタの値を、第1オペランドsrc1と3bitの即値im:3との組み合わせ(src1+im:3)を用いて指定されたメモリに格納するストア命令を指定している。

【0049】上位1bitが『0』であり、下位2bitが『10』である場合、その組み合わせは第1オペランドsrc1と3bitの即値との組み合わせ(src1+im:3)を用いたアドレッシングモードによりメモリから値を読み出し、格納先dst\_REGに指定されたレジスタに格納するロード命令を指定している。図11は、8bit、11bitの即値の指定が可能な比較命令／演算命令／分岐命令の命令フォーマットを示す。本命令フォーマットにおいて即値が8bitである場合は、5bitから7bitまでに第1オペランド及び演算結果の格納に相当するレジスタの指定が可能であり、8bitから15bitまでを即値の格納に用いる。即値が11bitである場合は、5bitから15bitまでを即値の格納に用いる。2bitから5bitまでの4bitを用いて下段の表のようにオペレーション内容が指定される。

【0050】下段の表は、当該4bitのうち、上位2bitと、下位2bitの組み合わせと、その組み合わせにて表現されるオペレーション内容とを示している。例えば上位2bitが『10』であり、下位2bitが『01』である場合、その組み合わせは第1オペランドで指定されたレジスタの値を第2オペランドと8bitの即値とを比較する比較命令を指定している。

【0051】上位2bitが『10』であり、下位2bitが『10』である場合、その組み合わせはオペランドと8bitの即値とを乗算する乗算命令を指定している。上位2bitが『11』であり、下位2bitが『11』である場合、その組み合わせは11bitの即値を分岐先アドレスとした絶対アドレス指定型の分岐命令を指定している。以上の説明において、オペレーション内容は一命令語長である16bitにて表現されるため、命令メモリ100内の如何なるオペレーション内容の命令を読み出す場合でも、またどの命令を解読・実行する場合でも、それらの読出、解読、実行が均一に一サイクル内に完遂することができる。即ち、命令の読出・解読・実行が完遂するまでの時間が命令間でバラツキが生じないように命令フォーマットが規定されているのである。特に図11において即値の幅を8bit、11bitとしていることに、この規定の意図は大きく現れている。つまり指定される即値を長くすると、命令語長が24bit、32bitまで拡張されてしまう恐れがあり、その拡張を防ぐため、命令語長が16bitに収まるように、即値長を8bit、11bitに留めているのである。一つの命令の実行に、どれだけのサイクル数が必要であるか

を示す単位(CPI(Cycle Per Instruction)という)にて表現すると、本I/OプロセッサのCPIは純粋に『1』であり、『命令を何回読み出したか』『命令を何回解読・実行したか』はサイクル数にて表現される。

【0052】命令読出回路10は、命令メモリ100内の読み出し先アドレスを命令メモリ100に出力する。命令解読制御部11(図中ではDecoder11)は、命令メモリ100が出力する命令を解読し、その解読結果に従って命令読出回路10及び演算実行部14を制御する。

【0053】レジスタセット12は、32bit長の汎用レジスタ(GR)と、16bit長の汎用レジスタとをそれぞれ24本有する。このように24本ものレジスタを有しているのは、32bit長のレジスタと、16bit長のレジスタとを4本ずつ、6本の非同期イベントタスクに割り当てるためである。このように各非同期イベントタスク毎に8本のレジスタを割り当てることにより、タスクの切り換え時にレジスタの値を退避・復元する必要がなくなる。

【0054】演算実行部14は、ALU、乗算器、バレルシフタを備え、命令解読制御部11による制御に基づきレジスタセット12における汎用レジスタの格納値を用いて演算を行う。非同期イベントタスクから別タスクへの切り換えられた際、演算実行部14はそれに対応する4本の32bit長の汎用レジスタ、4本の16bit長の汎用レジスタを用いて、切り換え後のタスクに含まれている命令についての演算を実行するタスク管理部15は、命令解読制御部11における命令実行を監視し、各タスクに割り当てられたスレッドの時間長が経過すれば次にスレッドを割り当てるべきタスクの識別子を命令読出回路10に出力する。

(4.1) 命令読出回路10は、読出先アドレスの更新をどのように行うか

命令読出回路10がどのような内部構成にて読出先アドレスの更新を行うかを図4を参照しながら説明する。命令読出回路10の内部構成を図4に示す。

【0055】図4に示すように、命令読出回路10はIF1+1保持部20、increment回路21、IF2保持部22、DECPC保持部23、タスク別PC格納部24、セレクト25及びセレクト26からなり、解読ステージの前に二段の読出ステージを行うパイプライン処理を実現するよう構成されている。この二段の読出ステージにおけるそれぞれの読出先アドレスを第1読出先アドレスIF1、第2読出先アドレスIF2という。

【0056】図4においてIF1はセレクト26が出力したアドレスをいい、IF2はIF2保持部22が保持するアドレスをいう。図12は、命令読出回路10内で行われるパイプライン処理を示すタイミングチャートである。以降、命令読出回路10の構成要素について言及する場合は、本タイミングチャートを引用するものとする。IF1+1保持部20は、カウント値を保持するためのカウント

値レジスタとして用いられるものであり、タスク別PC格納部24から何れか一つの読出先アドレスが出力されると、当該アドレスをカウント初期値として保持し、そのカウント値のインクリメントがincrement回路21より行われると、インクリメント後のアドレスを最新のカウント値として保持する。

【0057】この際、increment回路21より新たにインクリメントされたアドレスが出力されてくると、IF1+1保持部20はそれまで保持していたアドレスを信号線④を介してセレクト25に出力すると共にセレクト26に出力する。increment回路21は、セレクト26により出力されたIF1をクロック信号に従ってインクリメントする。インクリメントされたアドレスはIF1+1保持部20により保持される。セレクト26により出力されたIF1は、それまでIF1+1保持部20により保持されていた値であるので、increment回路21によるインクリメントにより、IF1+1保持部20が保持していたカウント値は一クロック毎にインクリメントされてゆく。

【0058】図12のタイミングチャートにおいて、読出先アドレスとして出力された命令0-0の読出先アドレスは、increment回路21によってインクリメントされる(図中のincl, inc2, inc3参照)。これによりIF1+1保持部20が保持する読出先アドレスは、命令0-1のアドレス、命令0-2のアドレス、命令0-3のアドレスというように更新されてゆく。

【0059】IF2保持部22は、クロック信号に同期して前回IF1+1保持部20により出力されたアドレスを命令を読み出すべき第2の読出先アドレスとして保持する。図12のタイミングチャートでは、第2の読出先アドレスとして命令0-0のアドレス、命令0-1のアドレス、命令0-2のアドレス、命令0-3のアドレスが出力されていることがわかる。これらのアドレスは第1の読出先アドレスより一サイクルだけ遅れていることがわかる。新たなアドレスがIF1+1保持部20から出力されると、IF2保持部22はそれまで保持していたアドレスをDECPC保持部23に出力し、信号線①を介してセレクト25にも出力する。出力後、IF1+1保持部20により出力されたその新たなアドレスを保持する。

【0060】DECPC保持部23は、前回IF2保持部22により出力されたアドレスを保持する。このアドレスは命令解読制御部11による解読制御の対象となる命令のアドレスと一致する。新たなアドレスがIF2保持部22により出力されると、その新たなアドレスを保持して解読対象のアドレスの更新を行う。DECPC保持部23が保持するアドレスは、IF2保持部22がそれまで保持していたアドレスであり、IF2保持部22が保持するアドレスは、IF1+1保持部20がそれまで保持していたアドレスであるから、IF1+1保持部20が保持しているアドレスと比較すると、DECPC保持部23に保持されているアドレスは二命令遅れであり、IF2保持部22が保持してい

るアドレスと比較すると、DECPC保持部23に保持されているアドレスは一命令遅れである。

【0061】タスク別PC格納部24は、タスク毎の読出先アドレスを保持する領域をその内部に有するメモリ回路或はタスク毎に設けられたアドレスレジスタである。内部領域或は個々のアドレスレジスタにおいて、個々に記憶されている読出先アドレスには、3ビット長のタスク識別子がアドレスとして付されている。尚、タスク(0)の読出先アドレスをIPC0と呼び、タスク(1)の読出先アドレスをIPC1と呼ぶ。以降タスク(2)、タスク(3)、タスク(4)、タスク(5)についても同様である。

【0062】タスク別PC格納部24におけるこれらの読出先アドレスの読み書き動作はタスク管理部15からタスクを切り換える旨の指示(タスク切替信号chg\_task\_exのHigh値)が出力された場合に行われる。このタスク切替信号chg\_task\_exがHigh値である場合においてタスク管理部15よりタスク切替信号chg\_task\_exと共に実行が済んだタスクの3ビット長の識別子(書き込みアドレス選択信号taskid(wr\_adr))及び次に実行すべきタスクの3ビット長の識別子(読み出しアドレス選択信号nxttaskid(rd\_adr))が出力されるが、タスク別PC格納部24はこれらの信号に従った内部領域の読み書きを行う。即ちタスク別PC格納部24は書き込みアドレス選択信号taskid(wr\_adr)をタスク別PC格納部24内のアドレスとして解釈して、それに指示される内部領域にセレクト25から出力された読出先アドレスを格納する。図12のタイミングチャートにおいてタスク切替信号chg\_task\_exが立ち上がったタイミングa1, b1において読出先アドレスの書き込みa2, b2が行われる。

【0063】書き込みa2が行われた時期には、IF2保持部22は命令0-3のアドレスを保持しており、IF1+1保持部20はこの命令0-3のアドレスに『1』を加算したアドレス0-4を信号線④を介してセレクト25に出力している。このように出力されると、アドレス0-4がタスク別PC格納部24に格納される。これにより、タスク別PC格納部24内のタスク(0)に次のスレッドが巡ってくれば、タスク(0)の読出はアドレス0-4から行われる。

【0064】書き込みb2が行われた時期には、IF2保持部22は命令1-3のアドレスを保持しており、IF1+1保持部20はこの命令1-3のアドレスに『1』を加算したアドレス1-4を信号線④を介してセレクト25に出力している。このように出力すると、インクリメントされたアドレス1-4がタスク別PC格納部24に格納される。これにより、タスク別PC格納部24内のタスク(1)に次のスレッドが巡ってくれば、タスク(1)の読出はアドレス1-4から行われる。

【0065】また読み出しアドレス選択信号nxttaskid(rd\_adr)がタスク管理部15より与えられると、タスク別PC格納部24は、その識別子をタスク別PC格納部24内のアドレスとして解釈して、それにより指示される内

部領域から非同期イベントタスクの読出先アドレスを取り出してセクタ26に出力する。図12のタイミングチャートにおいて、読み出しアドレス選択信号nxttaskidとして識別子(1)がタスク管理部15より与えられ、chg\_task\_exが立ち上がると(参照符号a3参照)、タスク別PC格納部24は、その識別子により指示されるタスク(1)をタスク別PC格納部24内のアドレスとして解釈して、そこに格納されている読出先アドレス(1-0)を取り出してセクタ26に出力する(参照符号a4参照)。

【0066】命令2-0のアドレスが読出先アドレスIF1として出力されたタイミングで、読み出しアドレス選択信号nxttaskidとして識別子(2)がタスク管理部15より与えられ、chg\_task\_exが立ち上がると(参照符号a5参照)、タスク別PC格納部24は、その識別子により指示されるタスク(2)の読出先アドレスとして読出先アドレス(2-0)を取り出してセクタ26に出力する(参照符号a6参照)。

【0067】以上のように読出先アドレスが出力されると、タスク(0)の命令は4命令だけ実行され、これに続いてタスク(1)の命令が4命令だけ実行される。以降、タスク(2)、タスク(3)、タスク(4)、タスク(5)の命令列がそれぞれ四命令ずつ実行されてゆく。4入力-1出力のセクタであるセクタ25は、①、②、③、④の信号線に伝送されるアドレスの何れかを選択的にタスク別PC格納部24に出力する。これらの①~④の何れを選択するかは図5に示す出力論理表に示す通りであり、命令解読制御部11が命令の解読結果に応じて出力するselpc信号に基づいてなされる。ここで④の信号線はIF1+1保持部20の出力であり、タスク管理部15よりタスク切替信号chg\_task\_exがHigh値に切り換えられた場合はセクタ25はこれを選択する。

【0068】また②、③の信号線はそれぞれ命令解読制御部11及び演算実行部14を接続されている。②は、絶対アドレス指定を用いた分岐命令の解読時において、分岐命令の即値を分岐先アドレスとして命令解読制御部11から取得するために選択される。③は、間接参照指定を用いた分岐命令の解読時において、演算実行部14により算出されたアドレスを分岐先アドレスとして用いる際に選択される。

【0069】セクタ26は、2入力-1出力のセクタであり、タスク切替信号chg\_task\_exがLowの期間においてincrement回路21によるインクリメント後のアドレスを選択してIF1としてincrement回路21及びIF2保持部22に出力する。タスク切替信号chg\_task\_exがHighの期間においてタスク別PC格納部24に格納されている次に実行すべきタスクのアドレスをIF1としてincrement回路21及びIF2保持部22に出力する。

【0070】以上のように構成された命令読み出し回路によれば、タスク切り換え時において、これまで実行されていたタスクの読出先アドレスはセクタ25により

タスク別PC格納部24に格納され、これに代えてタスク別PC格納部24に格納されている次のタスクのアドレスがセクタ26によりIF1として出力される。以上の読出先アドレスの切り換えは、一サイクルにて行われ、この読出先アドレスの切り換えにおいて、1/0プロセッサは余計な処理を行う必要がない。割り込み信号の通知時に行われる分岐のように先行的に読み出したアドレスを無効化することもない。従って、命令読出回路10による読出先アドレスの退避・復元は、オーバーヘッドの発生無しに行われる。

(4.2) タスク管理部15は、次に実行すべき非同期イベントタスクをどのように決定するか  
タスク管理部15がどのような内部構成にて次に実行すべき非同期イベントタスクを決定するかを図6に示すタスク管理部15の内部構成を参照しながら説明する。

【0071】図6は、タスク管理部15の内部構成を示す図である。図6に示すようにタスク管理部15はスレッドマネージャー61及びスケジューラ62からなり、スレッドマネージャー61はフリップフロップ51、カウンタ52、及び比較器54から構成され、スケジューラ62はタスクID保持部71、タスクID保持部72、タスクラウンド管理部73、及びプライオリティエンコーダ74から構成される。

【0072】フリップフロップ51は、次に実行される命令が何番目であるかを示す整数値(これをカウンタ値iと呼ぶ)を保持する。カウンタ52は、初期値が『1』に設定され、上限値が『4』に設定されたカウンタであり、クロック信号に同期してフリップフロップ51に保持されているカウンタ値を1,2,3,4,1,2,3,4というようにカウントアップする。

【0073】比較器54は、カウンタ52がカウントアップした値と、整数値『4』との比較を行い、もしカウンタ52のカウント値と整数値『4』とが一致すれば、タスク切替信号chg\_task\_exをHigh値にしてセクタ26及びタスク別PC格納部24に出力する。このようにカウンタ52が『4』をカウントする度にタスク切替信号chg\_task\_exがHighになることにより、セクタ26によるタスク別PC格納部24の選択は、4サイクルにつき一回行われることになる。この際、タスク別PC格納部24は次にスレッドを割り当てるべきタスクの読出先アドレスをセクタ26に出力しているから、タスク別PC格納部24からの読出先アドレスの出力は、4サイクルの実行につき一回行われることになる。

【0074】タスクID保持部71は、命令実行のカウントがカウンタ52により行われているタスクの識別子(これをタスク識別子taskidと呼ぶ。)を保持する。タスク識別子の一例を図7(a)に示す。図7(a)に示すように、タスク識別子は3ビットを用いて、命令メモリ100に記憶されている6つのタスクの何れか一つを指示する。命令実行のカウントが4回行われると、スケ

10

20

30

40

50

ジョーラ62においてプライオリティエンコーダ74から次にスレッドを割り当てるべきタスクのタスク識別子taskidが出力されてくるが、この際タスクID保持部71は、これまで保持していたタスク識別子taskidをタスクラウンド管理部73に出力し、新たに出力されてくるタスク識別子taskidを保持する。

【0075】タスクラウンド管理部73は、タスクID保持部71においてタスク識別子がどのように更新されてきたかを監視し、6ビット長のレジスタを用いて六個のタスクのうちスレッドが割り当てられたものを管理する。またカウンタ52が4命令の実行をカウントすると、タスクID保持部71によりタスク識別子taskidが出力され、タスクラウンド管理部73はスレッドの割り当てが済んだタスクを示す数値（この数値をラウンド値tasknと呼ぶ）をプライオリティエンコーダ74に出力する。ラウンド値tasknの一例を図7（b）に示す。このラウンド値tasknにおいて、第1ビットが「1」であれば、タスク(1)にスレッドが割り当てられたことを示し、第1ビットが「0」であればタスク(1)がそうでないことを示す。タスク管理レジスタ13の第2ビットが「1」であれば、タスク(2)にスレッドが割り当てられたことを示し、第2ビットが「0」であればタスク(2)がそうでないことを示す。

【0076】タスク(0)、タスク(1)にスレッドの割り当てが済めば、タスクラウンド管理部73は「000011」をプライオリティエンコーダ74に出力する。またタスク(0)、タスク(1)、タスク(2)にスレッドの割り当てが済めば、タスクラウンド管理部73は「000111」をプライオリティエンコーダ74に出力し、タスク(0)、タスク(1)、タスク(2)、タスク(3)にスレッドの割り当てが済めば、タスクラウンド管理部73は「001111」をプライオリティエンコーダ74に出力する。

【0077】尚タスク(0)～タスク(5)へのスレッドの割り当てが一巡すると、「111111」をプライオリティエンコーダ74に出力した後6ビット長のレジスタを「000000」にリセットする。プライオリティエンコーダ74は、タスクラウンド管理部73が出力したラウンド値tasknを受け取り、受け取ったタスクラウンド管理部73のラウンド値tasknにおいて何ビット目に「1」から「0」への反転が生じているかを検出する。このように「0」への反転が生じたビットを検出すると、この反転が生じたビットに割り当てられているタスクのタスク識別子taskidをタスクID保持部71に出力する。出力されたタスク識別子は、アドレス選択信号nxttaskid(rd\_adr)としてタスク別PC格納部24に出力される。

【0078】例えばタスクラウンド管理部73からラウンド値tasknとして「000011」が出力されると、プライオリティエンコーダ74はこのラウンド値tasknにおいて下位から第2ビット目に「0」への反転が生じていることを検出する（尚、最下位ビットをゼロビットと数え

ていることは留意されたい。）。この第2ビット目は図7（b）においてタスク(2)に割り当てられているビットであるから、プライオリティエンコーダ74はタスク(2)のタスク識別子taskidをタスクID保持部71に出力する。

【0079】例えばタスクラウンド管理部73からラウンド値tasknとして「000111」が出力されると、プライオリティエンコーダ74はこのラウンド値tasknにおいて下位から第3ビット目に「0」への反転が生じていることを検出する。この第3ビット目は図7（b）においてタスク(3)に割り当てられているビットであるから、プライオリティエンコーダ74はタスク(3)のタスク識別子taskidをタスクID保持部71に出力する。ラウンド値tasknは既にスレッドの割り当てが済んだタスクを示すから、このラウンド値tasknにおいて「0」へと反転しているビットは次にスレッドを割り当てるべきタスクがどれであるかを意味する。「0」へと反転しているビットをタスク識別子taskidに変換すると、タスクID保持部71には次にスレッドを割り当てるべきタスクが格納される。

【0080】タスクID保持部72は、プライオリティエンコーダ74が新たなタスク識別子taskidを出力すると、直前に出力されていたタスク識別子taskidを書き込みアドレス選択信号taskid(wr\_adr)として保持して、タスク別PC格納部24に出力する。以上のように構成された第1実施形態のI/Oプロセッサによって、どのようにスレッドが割り当てられるかを図13に示す。図13は、タスク(0)からタスク(5)までのタスクに割り当てたスレッドを時間順に並べることににより形成されるフレームという単位を示す。

【0081】タスク(0)～タスク(5)にはそれぞれ4サイクルのスレッドが割り当てられるので、24サイクルのフレームが形成されることになる。このフレームにより、図7に示した命令列は四命令ずつ均一に実行されることになる。

#### (5.) 動作クロック数の計算について

第1実施形態のI/Oプロセッサにおいて動作クロック数がどのように計算されるかについて図55を参照しながら説明する。図55は第1実施形態におけるスレッド割り当てにてビデオアウトタスク、オーディオアウトタスクを実行する場合に動画ストリーム、オーディオストリームがどのように処理されるかを示すタイミングチャートである。ここでビデオアウトタスク、オーディオアウトタスクのリアルタイム性を保証するには、動画ストリームを映像信号に変換するためのビデオアウトタスクはディスプレイの水平同期信号の周期に基づいた50μsecにおいて、一ライン分の画像を処理させねばならないとし、オーディオストリームを音声信号に変換するためのオーディオアウトタスクは90μsecという周期において、所定の音声ストリームを処理せねばならないとす



る。

【0082】この場合、先ず第1に上記の周期において『これだけの数の命令は最低実行しておかねばならない』という最低消化数を導出する。ここで一ライン分の映像データを復号するには、ビデオアウトタスク内の448個もの命令実行を最低消化せねばならず、一周期分の音声データを復号するには、オーディオアウトタスク内の808個もの命令実行を最低消化せねばならない。また、本実施形態においてCPI=1であり、命令数はサイ

クル数と等価であるため、最低消化数からビデオアウトタスク、オーディオアウトタスクに割り当てべきサイクル数が448(=4×112)サイクル、808(=4×202)サイクルと決定される。

【0083】またタスク(0)～タスク(5)はそれぞれを四サイクルずつ実行される。ここで命令読出回路10内でのセレクト25、セレクト26によるプログラムカウンタの切り換えによりオーバーヘッドは発生しないので、タスク(0)～タスク(5)の一巡に要する時間は実質24サイクルとなる。命令サイクル数は4であり、タスク総数は6であるので、全てのタスクは、24サイクルにて4命令ずつ実行されることになる。

【0084】そうすると、1サイクルに要する時間S1,S2は以下の式を満たさねばならない。

$$50\mu\text{sec} > 6 \times 4 \times 112 \times S1\text{nsec}$$

$$90\mu\text{sec} > 6 \times 4 \times 202 \times S2\text{nsec}$$

$$S1 < 50\mu\text{sec} / 6 \times 4 \times 112 = 18.6\text{nsec}$$

$$S2 < 90\mu\text{sec} / 6 \times 4 \times 202 = 18.56\text{nsec}$$

動作クロック数を1÷S2nsecの計算から定めると、

$$54\text{MHz} = 1 \div 18.56\text{nsec}$$

となる。以上の計算により、ビデオアウトタスク、オーディオアウトタスクの処理完遂に最適な動作クロック数を一義的に導出することができる。以上のように本実施形態によれば、動作クロック数の最適な下限値を所定の周期において実行せねばならない命令の最低消化数と、実行周期と、タスク総数とから一義的に導出することができる。動作クロックの最適な下限値が求まるので、動作クロック信号を一段と高いものに定めずとも、オーディオアウトタスク、ビデオアウトタスクのリアルタイム性を保証でき、動作クロックが低速なタイプが多い民生機器に搭載される場合でも、MPEGストリームを好適に復号することができる。

【0085】また、複数の非同期イベントタスクを時分割多重にて実行するI/Oプロセッサを設けることにより、割込にて非同期イベント処理を処理しなくて済むので、setup部104、VLD部105、IQ/IDCT部106は逆量子化、逆離散余弦変換、動き補償の処理に専念することができる。また、I/Oプロセッサはプログラムカウンタの値を回避し、復元するためのオーバーヘッドの発生無しに高速にタスクを切り換えて実行するので、動作クロックを高速にしなくても、MPEGストリームの再生に

要求されるリアルタイム性を満たすことができる。

【0086】尚、本実施形態において各タスクについてのスレッドを一律に4サイクルとしたが、各タスクの処理内容に応じて、それぞれ異なる値に設定しても良い。タスク別にサイクル数を可変にする場合、図6に示したスレッドマネージャーを図56に示すように構成する。本図においてセレクト200は、タスク(0)からタスク(5)までの固有のサイクル数である『cycle number for task0』～『cycle number for task5』と接続され、スケジューラ62が出力したnxttaskidに応じて対応するサイクル数を択一式に比較器54に出力する。図26において次にスレッドを割り当てべきタスクがタスク(0)なら、セレクト200は『cyclenumber for task0』を比較器54に出力する。このように構成すれば、タスク(0)からタスク(5)までのそれぞれの処理負荷に応じた最適なサイクル数を各タスクに分配することができる。またこのような最適なサイクル数をメモリ又はレジスタに記憶させておき、これを後日書き換えてもよい。

【0087】更に、本実施形態において実行すべきタスクを非同期イベントタスクに絞って説明したが、別のタスクでもよい。

(第2実施形態)第1実施形態では各タスクのスレッドに割り当てる命令数を一律四サイクルとしていたが、第2実施形態ではどれだけのサイクルを1スレッドに割り当てるかを、各タスクが独自に変更できるようにした実施形態である。スレッドへの割り当ては、Wait\_Until\_Next命令によりなされる。Wait\_Until\_Next命令とはスレッドにおける命令を途中で打ち切り、Wait\_Until\_Next命令の次順に位置する一連の命令列を次のスレッドに先送りする旨の命令である。尚Wait\_Until\_Next命令の解釈時における読み出し先アドレスの格納であるが、この際図4に示した5入力1出力のセレクト25は、①の信号線を介してIF1+1保持部20の出力を選択してタスクPC格納部24に格納する。

【0088】図14は、Wait\_Until\_Next命令(図中のWUN命令)を命令0-1としてその内部に有した非同期イベントタスクの一例であり、図16は、図14の非同期イベントタスクがどのように実行されるかを示すタイミングチャートである。Wait\_Until\_Next命令が図14に示すように命令0-1として非同期イベントタスクの第1番目の命令として格納されている場合、この命令0-1がDEC PC保持部23に格納され、命令解釈制御部11により解釈されると、セレクト25は、①の信号線を介して命令0-2の読出先アドレスをタスク別PC格納部24に格納する(図16の①も同様である。)。その後命令解釈制御部11は、命令0-2、命令0-3を無効化する(図16における『NOP』)。

【0089】ここで命令0-2は、I/Oプロセッサのローカルメモリ内のアドレスnem1の値をレジスタセット12内のレジスタr0に読み出す命令であり、命令0-3はI/Oプロ



セッサのローカルメモリ内のアドレスbuf1の値をレジスタr1に読み出す命令である。命令0-4はレジスタr0の値とレジスタr1の値とを乗じて、その結果をレジスタr2に格納する命令であり、命令0-5は、レジスタr2の値をI/Oプロセッサのローカルメモリ内のアドレスmem1に格納する命令である。

【0090】以上の四つの命令は、読出先アドレス、書込先アドレスが共にI/Oプロセッサのローカルメモリであり、命令0-0がWait\_Until\_Next命令であることにより、I/Oプロセッサのローカルメモリを読出先アドレス—書込先アドレスとした命令が一つのスレッドに納められたことがわかる。図15は、Wait\_Until\_Next命令の命令フォーマットの一例を示す。図15において、本命令は図9に示したレジスタ指定型の算術演算命令の命令フォーマットを有する。本フォーマットの11bitから13bitを「010」に指定することにより、Wait\_Until\_Next命令のオペレーションをI/Oプロセッサは実行する。

【0091】以上のように構成された第2実施形態のI/Oプロセッサによって、どのようにスレッドが割り当てられるかを図17を参照しながら説明する。タスク(1)、タスク(2)、タスク(3)、タスク(4)、タスク(5)には四サイクルのスレッドが割り当てられる。しかしタスク(0)には、wait\_until\_next命令が存在しているので、タスク(0)に割り当てられるスレッドは、wait\_until\_next命令が配置されている位置までとなり、タスク(0)に割り当てられるスレッドは2サイクルとなる。

【0092】タスク(0)～タスク(5)が一巡した後に形成される次のフレームにおいて、タスク(1)は、他のタスクと同様4サイクルのスレッドが割り当てられる。以上のように本実施形態によれば、同一メモリをアクセス先としたメモリアクセス命令を一つのスレッドにおいて集中して実行することができる。

(第3実施形態)第1実施形態においては、非同期イベントタスクを起動すべき事象が成立したか否かに係らず、6本の非同期イベントタスクに公平にスレッドを割り当てていたため、起動する必要がある非同期イベントタスクと、その必要がある非同期イベントタスクとが実質同時であるという悪平等が横行していた。第3実施形態は、非同期イベントタスクを起動すべき事象が成立しているか否かのチェックを非同期イベントタスクに行わせ、起動すべき事象が未成立のタスクについては、サイクルの割当数を削減するように制御を行う。

【0093】第3実施形態におけるI/Oプロセッサの内部構成を図18に示す。図18においてI/Oプロセッサが命令メモリ100、命令読出回路10を備えている点は第1実施形態と変わりはない。第3実施形態において新規なのは、命令解釈制御部11及び演算実行部14がそれぞれ、命令解釈制御部81及び演算実行部84に置き換えられている点である。また7本の状態監視レジスタが追加され、タスク管理部15の内部にタスク管理レ

ジスタ13が備えられた点である。

【0094】七本の状態監視レジスタCR1～CR7は、それぞれが5ビット長のレジスタであり、AVデコーダ内部においてI/Oプロセッサの周辺に位置する構成要素、即ち、ビデオ出力部108、バッファメモリコントローラ110、RAMコントローラ111、FIFOコントローラ112等の構成要素と接続され、I/Oプロセッサの周辺部における5個の事象の成否が各ビットに反映されている(尚、ビデオ出力部108、バッファメモリコントローラ110、RAMコントローラ111、FIFOコントローラ112との接続は複雑であるため、その図示は省略している。)

【0095】その一例として状態監視レジスタCR1、状態監視レジスタCR2、状態監視レジスタCR3について図23を参照しながら説明する。図23は、状態監視レジスタCR1、状態監視レジスタCR2、状態監視レジスタCR3のビット構成を示した図である。状態監視レジスタCR1の第0ビットは、通常は「0」に設定され、バッファメモリバス121においてバッファメモリ102へのデータ転送が所定回数が行われたときのみ「1」に設定される。

【0096】状態監視レジスタCR1の第1ビットは、通常は「0」に設定され、バッファメモリバス121に出力されたMPEGストリームのスタートコードが検出されたときのみ「1」に設定される。状態監視レジスタCR1の第2ビットは、通常は「0」に設定され、バッファメモリバス121に出力されたMPEGストリームに1バイト以上の有効データの存在が確認されたときのみ「1」に設定される。

【0097】状態監視レジスタCR1の第3ビットは、通常は「0」に設定され、バッファメモリ102に、バッファに有効ビットが3Byte以上存在するときのみ「1」に設定される。また本ビットは、パージングタスクと、オーディオストリーム転送制御タスク、動画ストリーム転送制御タスクとの間の通信に用いられる。状態監視レジスタCR2の第0ビットは、通常は「0」に設定され、バッファメモリ102上でSDRAM300への転送をサポートする旨の要求が発せられれば「1」に設定される。また本ビットは、パージングタスクと、オーディオストリーム転送制御タスク、動画ストリーム転送制御タスクとの間の通信に用いられる。

【0098】状態監視レジスタCR2の第1ビットは、通常は「0」に設定され、DMA転送が完了したときのみ「1」に設定される。状態監視レジスタCR2の第2ビットは、通常は「0」に設定され、ビットストリームFIFO103に対しての転送要求がある場合のみ「1」に設定される。状態監視レジスタCR2の第3ビットは、通常は「0」に設定され、FIFOコントローラ112内部の制御レジスタが更新されれば、「1」に設定される。状態監視レジスタCR3の第0ビットは、通常は「0」に設定さ

れ、バッファメモリ 102 が DMA\_READY 状態となれば『1』に設定される。

【0099】状態監視レジスタ CR3 の第 1 ビットは、通常は『0』に設定され、SDRAM 300 への DMA 転送が完了すれば『1』に設定される。状態監視レジスタ CR3 の第 2 ビットは、通常は『0』に設定され、外部に接続されたディスプレイ装置における水平帰線信号の立ち下がりにて『1』に設定される。

【0100】状態監視レジスタ CR3 の第 3 ビットは、通常は『0』に設定され、外部に接続されたディスプレイ装置におけるビデオ信号が水平ブランキング期間になると『1』に設定される。状態監視レジスタ CR3 の第 4 ビットは、通常は『0』に設定され、ビデオアウトタスクに対する処理要求が発生した場合のみ『1』に設定される。

【0101】以上のように 3 本の状態監視レジスタは、AV デコーダ内の各バッファやコントローラの制御状態を表している。他の状態監視レジスタも同様であり、7 本の状態監視レジスタにより 1/0 プロセッサは、各非同期イベントタスクは、自身の起動要因となる各種事象が成立しているか否かを監視することができる。演算実行部 84 は、演算実行部 14 と同様の機能を有した構成である。演算実行部 14 と比較して第 1 に新規なのは命令解読制御部 81 から CR1~CR7 といった各状態監視レジスタに付された番号と即値とを受け取り、指定された状態監視レジスタの保持値と受け取った即値との比較のための減算を行う点である。この減算結果に応じてゼロフラグ、キャリーフラグをオン/オフして状態監視レジスタの保持値と即値との一致、不一致、大小を命令解読制御部 81 に知らせる。

【0102】上記の通知において命令解読制御部 81 は現在実行中の演算を無効化する旨の信号を出力し得るが、これを受け取ると演算実行部 84 は汎用レジスタへの値の格納を中断する。命令解読制御部 81 は、命令解読制御部 11 と同様の機能を有した構成であるが、命令解読制御部 11 と異なるのは命令解読制御部 81 は事象待ちループを形成し得る命令を命令メモリ 100 から読み出して、これを解読した際に CR1~CR7 といった各状態監視レジスタに付された番号と即値とを演算実行部 84 に出力する。尚、事象待ちループを構成する命令とは、以下の {例 1} の旨の内容である。状態監視レジスタの指定と即値とを演算実行部 84 に出力した後、演算実行部 84 がゼロフラグ、キャリーフラグにより保持値と即値との一致、不一致、大小を命令解読制御部 81 に知らせると、命令解読制御部 81 はこれを参照し、もし保持値と即値との不一致が通知されれば、タスク管理レジスタ 13 に事象不成立を示す信号を出力し、この次の命令の実行結果を無効化する旨の信号を演算実行部 84 に出力して読出先アドレスの値を先に進めないよう、命令読出回路 10 に通知する。もし保持値と即値との不一致が

通知されれば、タスク管理レジスタ 13 に事象成立を示す信号を出力する。

{例 1}

命令

事象 i の成立の真偽を判定し、事象 i の成立が偽ならば読出先アドレスを先には進めない。

【0103】また本命令のニーモニックを {例 2} に示す。

{例 2}

10 cmp\_and\_wait cr\_statei, 即値

{例 2} において第 1 オペランドに cr\_statei (i=0,1,2,3,4,...7) が記述されているのは、1/0 プロセッサにおける 7 本の状態監視レジスタのうち何れかを指示できることを意味する。

【0104】この第 1 オペランドにおける状態監視レジスタの指定と、第 2 オペランドにおける即値指定とを組み合わせ、35 もの事象のうち、任意のものの成否を確認することができる。上記命令を以降の説明では Cmp\_And\_Wait 命令と呼ぶ。また上述した事象待ちループは、この Cmp\_And\_Wait 命令の第 1 オペランド、第 2 オペランドにおいて指定された事象の成否が何度も否と判定されることにより発生することはいうまでもない。

【0105】第 3 実施形態における変更が加えられた命令読出回路 10 の構成を図 19 に示す。図 19 において命令読出回路 10 が IF1+1 保持部 20、increment 回路 21、IF2 保持部 22、DECPC 保持部 23、タスク別 PC 格納部 24、セレクト 25、セレクト 26 を備えている点は第 1 実施形態と差はない。命令読出回路 10 において新規なのは、DECPC 保持部 23 の出力段に接続されたフリップフロップ 27 である。

【0106】フリップフロップ 27 は、Cmp\_And\_Wait 命令が指定した事象が不成立である場合に備え、命令解読制御部 81 が解読中の命令の読出先アドレスをセレクト 25 に出力できるよう DECPC 保持部 23 の値を保持している。もしプログラムカウンタ値を戻す旨の指示が命令解読制御部 11 より与えられれば、フリップフロップ 27 はその保持値を ⑤ の信号線を通じてセレクト 25 に出力し、セレクト 25 は、このフリップフロップ 27 の保持値をタスク別 PC 格納部 24 に格納するよう、タスク別 PC 格納部 24 の入力元を ⑤ の信号線に切り換える。第 3 実施形態における命令読出回路の出力論理表を図 20 に示す。

【0107】図 26 は、Cmp\_and\_Wait 命令の解読時のフレームにおける 1/0 プロセッサの構成要素についてのタイミングチャートであり、本タイミングチャートにおけるフリップフロップ 27 の役割について説明する。図 26 においてフリップフロップ 27 が保持値として Cmp\_and\_Wait 命令のアドレスに相当する命令 0-0 のアドレスを参照符号 a7 に示すように出力すると、命令 0-0 のアドレスは IPC0 としてタスク別 PC 格納部 24 に格納される。こ

れによりタスク別PC格納部24には、IPC0としてCmp\_and\_Wait命令のアドレスが格納されることになる。

【0108】本実施形態におけるタスク管理部15の内部構成を図21に示す。図21に示すタスク管理レジスタ13は、事象待ちループ（以降wait状態と呼ぶ）に陥ったタスクを個別に管理するためのビットが割り当てられたレジスタである。タスク管理レジスタ13は命令解読制御部81から事象不成立を示す信号が通知されると、タスクID保持部72が保持しているタスク識別子taskidを参照し、そのタスク識別子taskidに対応するビットを『1』に切り換える。逆に事象成立を示す信号が通知されると、タスクID保持部72が保持しているタスク識別子taskidを参照し、そのタスク識別子taskidに対応するビットを『0』に切り換える。

【0109】図22は、タスク管理レジスタ13におけるwait状態タスクの管理用のビット構成を示す。本図において第0ビットが『1』であれば、タスク(0)がwait状態であることを示し、第0ビットが『0』であればタスク(0)がそうでないことを示す。また本図において第1ビットが『1』であれば、タスク(1)がwait状態であることを示し、第1ビットが『0』であればタスク(1)がそうでないことを示す。タスク管理レジスタ13の第2ビットが『1』であれば、タスク(2)がwait状態であることを示し、第2ビットが『0』であればタスク(2)がそうでないことを示す。タスク管理レジスタ13による事象待ちループの管理により、複数のタスクにおいて同時多発した事象待ちループを個別に管理することができる。

【0110】図26のタイミングチャートにおいて、命令0-0がCmp\_and\_Wait命令であると解釈され、事象が不成立であると判定されたものとする。この場合、命令解読制御部81はタスク(0)が事象成立待ち状態になった旨を参照符号a8に示すように通知する。そうすると、タスク管理レジスタ13はタスク(0)のタスク識別子に対応するビットを参照符号a9に示すように『1』に設定する。

【0111】続いて『事象成立待ち時』、『事象成立時』、『事象成立時以降』においてタスク管理レジスタ13に割り当てられた各ビットがどのように設定されるかについてタイミングチャートを参照しながら説明する。図27に事象成立待ち時、図28に事象成立時、図29に事象成立時以降におけるタイミングチャートを示す。

【0112】タスク(0)におけるCmp\_and\_Wait命令が成立を待っていた事象が図28において成立したとする。この場合、命令解読制御部81は事象が成立した旨を参照符号d1に示すように通知する。そうすると、タスク管理レジスタ13はタスク(0)のタスク識別子に対応するビットを参照符号d2に示すように『0』に設定する。図21を参照して、第3実施形態におけるスレッドマネージャ61の内部構成について説明する。スレッドマネ

ージャ61においてフリップフロップ51、カウンタ52、比較器54を備えている点は第1実施形態と差違は無い。図21において新規なのはIDコンバータ53及びセレクト55である。

【0113】IDコンバータ53は、タスクID保持部72に次にスレッドを割り当てるべきタスクの識別子が出力されると、タスク管理レジスタ13においてwait状態に設定されているタスクと次にスレッドを割り当てるべきタスク識別子taskidとが一致しているかを判定し、もし一致していればセレクト55にHigh値を出力し、不一致ならばセレクト55にLow値を出力する。

【0114】セレクト55は、IDコンバータ53が出力した信号のHigh値、Low値に応じて、『2』或は『4』の数値を択一式に比較器54に出力する。図26において次にスレッドを割り当てるべきタスクがタスク(0)なら、参照符号a10においてセレクト55にLow値が出力されるので参照符号a12に示すように、セレクト55は『4』を比較器54に出力する。しかし図26の参照符号a9においてタスク(0)についてのビットが『1』に設定されて、図27の参照符号g1に示すようにセレクト55にHigh値が出力されると、セレクト55は参照符号g2に示すように『2』を比較器54に出力する。

【0115】図28の参照符号d2においてタスク(0)についてのビットが『0』に設定されて、図29において参照符号g5に示すようにセレクト55にHigh値が出力されると、セレクト55は参照符号g6に示すように『4』を比較器54に出力する。セレクト55が『2』又は『4』の数値を択一的に比較器54に出力することにより、2サイクルのスレッド、4サイクルのスレッドの何れか一方が各タスクに割り当てられる。セレクト55が『2』を出力した際には、その第2番目の命令を無効化するので、その結果、1サイクルのスレッドが割り当てられることになる。

【0116】タイミングチャートの時間軸に沿って、図26から図29までのタイミングチャートに示されている命令読出回路10の構成要素及びタスク管理部15の構成要素の動作について説明する。初期状態においてタスク管理レジスタ13のビットはオールゼロであるものとする。そうして図26においてタスク(0)から命令の読み出しが開始されると、タスク(0)についてのビットはタスク管理レジスタ13において『0』に設定されているので、参照符号a12に示すようにセレクト55はカウンタの上限値として『4』を出力する。カウンタの上限値が『4』に設定されたので、タスク(0)を四サイクル実行しようとするが、命令0-0がCmp\_and\_Wait命令であるので、参照符号a7に示すようにタスク別PC格納部24に命令0-0の読出先アドレスを書き込む。それと共に、参照符号a8,a9に示すようにCmp\_and\_Wait命令の実行時においてタスク管理レジスタ13の第0ビットを『1』に設定する。この設定により、タスク(0)は事象成

立待ちとなる。このようにビットを設定した後、タスク(1)、タスク(2)、タスク(3)、タスク(4)、タスク(5)の実行が済んで次のフレームに移行したものとす

【0117】事象成立待ち状態においてタスク管理レジスタ13のビットはタスク(0)のみが『1』であり、その他のビットはゼロであるものとする。そうして図27においてタスク(0)から命令の読み出しが開始されると、タスク(0)についてのビットはタスク管理レジスタ13において『1』に設定されているので、参照符号g2に示すようにセクタ55はカウンタの上限値として『2』を出力する。カウンタの上限値が『2』に設定されたので、タスク(0)を2サイクル実行しようとするが、命令0-0がCmp\_and\_Wait命令であるので、タスク別PC格納部24に命令0-0の読出先アドレスを書き込む。このような書き込み後、タスク(1)、タスク(2)、タスク(3)、タスク(4)、タスク(5)の実行が済んで次のフレームに移行したものとす

【0118】次のフレームにおいて事象が成立したものとす。図28においてタスク(0)から命令の読み出しが開始されると、タスク(0)についてのビットはタスク管理レジスタ13において『1』に設定されているので、参照符号g4に示すようにセクタ55はカウンタの上限値として『2』を出力する。カウンタの上限値が『2』に設定されたので、タスク(0)を2サイクル実行しようとする

【0119】命令0-0であるCmp\_and\_Wait命令の実行時において、事象成立が確認されるので、参照符号d1,d2に示すようにCmp\_and\_Wait命令の実行時においてタスク管理レジスタ13の第0ビットを『0』に設定する。この設定により、タスク(0)は事象成立待ちから解放される。このようにビットを設定した後、タスク(1)、タスク(2)、タスク(3)、タスク(4)、タスク(5)の実行が済んで次のフレームに移行したものとす

【0120】図29に示す事象成立以降のフレームにおいては、タスク(0)についてのビットがタスク管理レジスタ13において『0』に設定されているので、参照符号g6に示すようにセクタ55はカウンタの上限値として『4』を出力する。カウンタの上限値が『4』に設定されたので、タスク(0)を4サイクル実行する。以上のように構成された第3実施形態のI/Oプロセッサによって、どのようにスレッドが割り当てられるかを図24(a)～図24(d)を参照しながら説明する

【0121】図24(a)においてタスク(0)、タスク(3)、タスク(4)、タスク(5)は何れもメモリ転送制御タスクであり、その第1番目の命令がCmp\_and\_Wait命令であるものとする。ここで図24(b)の状態では、これらのCmp\_and\_Wait命令が待機している事象が未成立であるものとする。タスク(0)、タスク(3)～タスク(5)に2サイクルのスレッドが割り当てられ、第2命令は無効化されているが、タスク(1)、タスク(2)には4サイクル

のスレッドが割り当てられる

【0122】図24(c)においてタスク(0)～タスク(5)が何回か巡回し、タスク(0)のCmp\_and\_Wait命令が待っていた事象のみが成立したものとす。そうすると当該事象が成立したフレームにおいては、タスク(3)～タスク(5)には依然として2サイクルのスレッドが割り当てられており、第2命令は無効化されているが、タスク(0)は事象が成立したので第2命令は実行される

【0123】図24(d)において事象j0の成立以降のフレームにおいては、タスク(0)には、タスク(1)、タスク(2)と同一の4サイクルのスレッドが割り当てられる。図25は、Cmp\_and\_Wait命令の命令フォーマットの一例を示す。図25において、5bitから7bitまでの3bitを用いて7本の状態監視レジスタのうち何れかを指定でき、11bitから15bitまでを用いて5bit長の即値を格納できる。本フォーマットの9bitから10bitを用いてオペレーション内容の指定が可能である。この2bitを『10』に指定することにより、Cmp\_and\_Wait命令のオペレーションをI/Oプロセッサは実行する。また、この2bitを『00』に指定することにより、七本の状態監視レジスタのうち、一本のものの値と5bit即値との比較命令のオペレーションをI/Oプロセッサは実行する

【0124】以上のように本実施形態によれば、メモリアクセスが可能な状態を待っているメモリ転送制御タスクにより短い時間長のスレッドを割り当てることにより、他のタスクの進捗を向上させることができる

(第4実施形態)第4実施形態は、ディスプレイの水平ブランキング期間、垂直ブランキング期間において、ビデオアウトタスクにより多くのスレッドを割り当てる技術に関する

【0125】スレッドの増加を行うI/Oプロセッサは、第1実施形態におけるタスク管理部15の構成に図30～図31に示す変更を加えて実現される。第4実施形態におけるI/Oプロセッサは、命令メモリ100、命令読出回路10、命令解釈制御部81、演算実行部84、タスク管理部15を備えている点は第3実施形態と変わりはない。第4実施形態において新規なのは、タスク管理レジスタ13が緊急状態遷移許可信号iexecmodeを監視している点である

【0126】第4実施形態におけるタスク管理部15の内部構成を図30に示す。図30におけるタスク管理レジスタ13における3ビットは、タスク管理レジスタ13内の3ビットを用いて緊急扱いを要するタスクの識別子が指定される。当該3ビットにおいて六個のタスクのうちどのタスクを緊急扱いするかは、上記緊急状態遷移許可信号iexecmodeがどのような内容の信号であるかに応じて適宜変更する。本実施形態では、緊急状態遷移許可信号iexecmodeは水平ブランキング期間、垂直ブランキング期間等に相当し、ビデオアウトタスクの識別子をタスク管理レジスタ13における3ビットに指定してい

る。

【0127】何れかのタスクを緊急扱いを要するタスクとしてタスク管理レジスタ13に登録するかは、何れのタスク内にemergency命令が含まれているかをタスク管理レジスタ13が参照することにより、決定されるものとする。命令解読制御部81が次に実行すべき命令を解読した際、その解読結果がemergency命令であれば、タスク管理レジスタ13は当該命令を含むタスクを緊急扱いを要するタスクとしてその識別子を保持する。

【0128】第4実施形態におけるスケジューラ62の内部構成を図31に示す。図31においてスケジューラ62がタスクID保持部72を備えている点は第1実施形態と差違は無い。図31において新規なのは、タスクラウンド管理部73及びプライオリティエンコーダ74がそれぞれ、タスクラウンド管理部75及びプライオリティエンコーダ80に置き換えられている点である。またコンバータ76、緊急タスクマスク部77及びタスクID保持部83が追加されている点である。

【0129】タスクラウンド管理部75がタスクラウンド管理部73と異なるのは、タスクラウンド管理部73がタスクID保持部71においてタスク識別子がどのように更新されてきたかを監視し、6ビット長のレジスタを用いて六個のタスクのうちスレッドの割り当てが済んだものを管理するのに対して、タスクラウンド管理部75は、スレッドの割り当てが済んだタスクの管理を、通常期間におけるタスク管理と、緊急期間における通常タスクの管理とに区別して行う点である。例えばタスク管理レジスタ13においてタスク(1)をインターリーブ式に実行する場合、6個のタスクのうち5個のタスクがタスク(0)、タスク(2)、タスク(3)、タスク(4)、タスク(5)というように巡回することになる。

【0130】5個のタスクを上記のように巡回する場合、実行の済んだタスクを示すラウンド値はタスク管理レジスタ13において管理されている緊急タスク抜きの数値で表現する必要がある。このようにタスクラウンド管理部75は緊急タスク抜きのラウンド値を管理するため、ラウンド値tasknとは別に緊急タスク抜きのラウンド値taskneを緊急タスクマスク部77に出力する。ちなみにtaskneの『ne』とは、『Not』及び『Emergency』の頭文字に由来している。

【0131】コンバータ76は、タスク管理レジスタ13に管理されている緊急タスクの識別子を6ビットのラウンド値emgcytaskに変換する。緊急タスクマスク部77は、コンバータ76により変換されたラウンド値emgcytaskと、タスクラウンド管理部75が出力したラウンド値taskneとの6ビット長の論理和をとるOR回路78を備える。

【0132】ここでOR回路78において論理和をとるのは、ラウンド値taskne及び緊急タスクを示すラウンド値の両者を含んだ6ビット長のラウンド値を算出すること

を意味する。例えばラウンド値taskneがタスク(0)、タスク(2)、タスク(3)にスレッドが割り当てられたことを示す『001101』であり、ラウンド値emgcytaskが緊急タスクがタスク(1)であることを示す『000010』である場合、OR回路78の出力は、タスク(0)、タスク(1)、タスク(2)、タスク(3)にスレッドが割り当てられたことを示す『001111』となる。

【0133】プライオリティエンコーダ80は、プライオリティエンコーダ74の機能に加えて、タスクラウンド管理部75が出力したラウンド値taskne(ラウンド値maskne)を受け取り、受け取ったタスクラウンド管理部73のラウンド値taskne(ラウンド値maskne)において何ビット目に『1』から『0』への反転が生じているかを検出する。このように『0』への反転が生じたビットを検出すると、この反転が生じたビットに割り当てられているタスクのタスク識別子taskidをそれぞれ個別にタスクID保持部83に格納させる。

【0134】例えば緊急タスクのラウンド値とラウンド値taskneとの論理和であるラウンド値maskne『000111』が緊急タスクマスク部77から出力されると、プライオリティエンコーダ80はこのラウンド値maskneにおいて下位から第3ビット目に『0』への反転が生じていることを検出する。この第3ビット目は図7(b)においてタスク(3)に割り当てられているビットであるから、プライオリティエンコーダ80はタスク(3)のタスク識別子taskidをセレクト82に出力する。

【0135】cstate記憶部85は、1/0プロセッサが通常状態であるか緊急状態であるかを示すフラグcstateを記憶する。図33は、cstate記憶部85が記憶するフラグの状態遷移図である。本図において、通常状態から緊急状態への遷移は、緊急状態遷移許可信号iexecmodeがHighであり、尚且つタスク切替信号chg\_task\_exがHigh値になった場合に行われる。逆に緊急状態から通常状態への遷移は、タスク切替信号chg\_task\_exがHigh値になった場合に行われる。図34は緊急状態遷移許可信号iexecmode、タスク切替信号chg\_task\_ex、cstate間のタイミングを示すタイミングチャートである。

【0136】iexecmode信号がHighに立ち上がったタイミングa30において参照符号a31に示すようにタスク切替信号chg\_task\_exがHigh値になったとする。そうするとcstateは参照符号a32に示すようにHighとなって緊急状態に遷移する。タスク切替信号chg\_task\_exが参照符号a33に示すように再度Highになるとcstateは参照符号a34に示すようにLow値となって通常状態に戻る。このような緊急状態から通常状態への遷移、通常状態から緊急状態への遷移を繰り返すことになる。

【0137】セレクト82は、入力されてきた緊急状態遷移許可信号iexecmodeがHighであり、尚且つcstateが参照符号a32に示すようにHighである場合、(cstateの逆の真理値cstate!が参照符号a35に示すようにLowであ

る場合)に、タスク管理レジスタ13に記憶されているタスク識別子taskidnを選択してタスクID保持部72に出力する。その他の場合、プライオリティエンコーダ80の出力であるラウンド値maskneから変換されたタスク識別子taskidneを選択してタスクID保持部72に出力する。図32は、セクタ82におけるtaskidnの真理値表を示す図である。

【0138】このようにラウンド値maskneから変換されたタスク識別子taskidneと、緊急タスクの識別子とをセクタ82が緊急状態遷移許可信号iexecmodeのHigh期間/Low期間に応じて切り換えて出力することは、緊急状態遷移許可信号iexecmodeのHigh期間/Low期間に応じて、緊急タスク抜きの何れかのタスクと、緊急タスクとを切り換えてタスクID保持部72に格納することを意味する。

【0139】例えばタスク(1)が緊急タスクであるものとする、緊急タスク抜きのタスク識別子として、タスク(0)、タスク(2)、タスク(3)、タスク(4)、タスク(5)のようにセクタ82はタスク識別子を出力する。通常状態では、図35(a)に示すように、セクタ82により緊急タスク抜きの何れかのタスクが出力されるため、タスク(0)、タスク(2)、タスク(3)、タスク(4)、タスク(5)のように、タスク(1)に割り当てられるスレッドは0サイクルとなる。

【0140】緊急状態では、図35(b)に示すように、セクタ82により緊急タスク抜きの何れかのタスクと、緊急タスクとが選択的に出力されるため、タスク(0)、タスク(1)、タスク(2)、タスク(1)、タスク(3)、タスク(1)、タスク(4)、タスク(1)、タスク(5)、タスク(1)のように、タスク(1)は二回に一回の割合でスレッドが割り当てられる。セクタ82による緊急状態におけるタスク識別子の出力では、タスク(1)にタスク実行が巡ってくるのは二回につき一回となり確率に換算すると1/2となる。

【0141】このような1/2の確率は、通常の実行確率の3倍という極めて高確率であり、緊急タスクは非常に高速に実行されることになる。以上のように本実施形態によれば、水平ブランキング期間、垂直ブランキング期間においてビデオアウトタスクのみを優先して実行することができる。この期間において動画データを映像信号に変換すれば、映像信号をディスプレイの表示期間までに好適に処理することができる。

【0142】尚第4実施形態の構成では、緊急タスクを優先して実行するため、緊急タスク以外のタスクの実行頻度が落ちてしまうという弊害が発生するが、これを防止する防止機構を設けても良い。その防止機構とは、

『最低限これだけの実行速度を維持しなければならない』という制約付きタスク(例外タスクという)の識別子をタスク管理レジスタに記憶させ、この例外タスクについては、四命令実行というタスク切り換えの条件を除

外する旨の信号をスレッドマネージャー61宛に出力させる。

【0143】スレッドマネージャー61内の比較器54は、この旨の信号を受け付けると、タスク切替信号の出力を所定時間だけ猶予する。ここでの所定時間の時間長とは緊急タスク優先による各タスクの遅れを取り戻せるだけの時間長である。このように所定時間だけタスク切替信号の出力を猶予することにより、例外タスクの処理を4命令以上行わせ、緊急タスク挿入に伴う処理の遅延を取り戻させることができる。

【0144】また本実施形態は、外部から入力されてくる特定信号に同期して、特定タスクを高速化に実行することについて述べているが、特定命令によって高速実行するタスクを定めても良い。状態監視レジスタCR2の第3ビットは、通常は『0』に設定され、外部に接続されたディスプレイ装置におけるビデオ信号が帰線期間になると『1』に設定されるので、状態監視レジスタCR2と、即値とをオペランドに指定し、これらの一致、不一致を判定させる旨のCMP命令をビデオアウトタスクに配置しておき、この命令についての演算の実行結果が真となる場合のみ、ビデオアウトタスクを高速に実行してもよい。命令のオペランドにて緊急扱いを要するタスクを指定できるようにしてもよい。

【0145】更に通常状態にて実行すべきタスクを緊急タスク抜きのラウンド値taskneにて指定したが、通常状態にて緊急タスクを実行してもよい。

(第5実施形態)第5実施形態は、タスク同士の連携により、タスクの巡回効率を向上させることを意図した実施形態である。

【0146】ここでタスク同士の連携とは、特定時期が到来しないとスレッドの割り当てが無駄に終わるタスクは自ら進んで休眠状態に入り、休眠状態の解除を他のタスクに委ねる。そして解除を委ねられたタスクは、その休眠状態を解除すべき時期が到来するのを待ち、もし到来すれば特定タスクの休眠状態を解除することをいう。

【0147】タスクの休眠は、複数のタスクにおいて同時多発する場合もあるし、全く発生しない場合もある。休眠状態に陥ったタスクを個別に管理できるよう、第5実施形態ではタスク管理レジスタ13に休眠(sleep状態)に入ったタスクを管理するためのビットを割り当てている。図36は第5実施形態におけるタスク管理部15の構成であり、図37はタスク管理レジスタ13のビット構成及び、これらの各ビットがどのようなフラグとして機能するかを示す説明図である。本図においてタスク管理レジスタ13の第0ビットが『0』であれば、タスク(0)をsleep状態として処理することを示し、第0ビットが『1』であればタスク(0)がそうでないことを示す。

【0148】タスク管理レジスタ13の第1ビットが『0』であれば、タスク(1)をsleep状態として処理する

10

20

30

40

50

ことを示し、第1ビットが『1』であればタスク(1)がそうでないことを示す。タスク管理レジスタ13の第2ビットが『0』であれば、タスク(2)をsleep状態として処理することを示し、第2ビットが『1』であればタスク(2)がそうでないことを示す。これらのビットをタスクが切り換えることにより、自身を休眠状態にしたり、他のタスクの休眠状態を解除したりする。

【0149】タスク管理レジスタ13のビット操作は、以下の{例3}{例4}に二モニクを示す命令によりなされる。

{例3}

sleep\_task

{例4}

wake\_task task(k)

{例3}において『sleep\_task命令』は、自身をsleep状態にする命令である。{例4}において『wake\_task命令』は、第1オペランドにより6つのタスクのうち何れか一つの識別子を指定できる。このような二種の命令を各タスク内に配置することにより、各タスクは休眠状態の移行と、これの解除とを交互に行う。尚、sleep\_task命令においては第1オペランドにより他のタスクを指定できるようにしてもよい。

【0150】図38は、第4実施形態におけるスケジューラ62の内部構成を示す図である。図38においてスケジューラ62がタスクID保持部71、タスクID保持部72、タスクラウンド管理部73、プライオリティエンコーダ74を備えている点は第1実施形態と差違は無い。本図において新規なのはインバータ94及びOR回路95からなるタスクスキップ管理回路93である。

【0151】タスクスキップ管理回路93は、タスク管理レジスタ13においてsleep状態に割り当てられている6ビットを反転するインバータ94と、反転された6ビットと、タスクラウンド管理部73から出力されたラウンド値tasknとの論理和をとるOR回路95からなる。ここでタスク(0)、タスク(1)、タスク(2)の実行が済んで『000111』というラウンド値tasknがタスクラウンド管理部73から出力され、またタスク管理レジスタ13においてタスク(3)がsleep状態に設定されているものとする。この場合タスク管理レジスタ13のsleepビットは『110111』となり、インバータ94によりこれの反転値『001000』が出力される。インバータ94による反転値『001000』とラウンド値taskn『000111』との論理和は、『001111』となる。この論理和がプライオリティエンコーダ74に出力されると、第4ビットにおいて『1』から『0』への反転が行われていることが検出され、タスク(4)の識別子がタスクID保持部71及びタスクID保持部72に出力される。タスク(4)の識別子が出力されると、タスク(0)、タスク(1)、タスク(2)、タスク(4)の順に実行が行われることになる。

【0152】図39は、sleep命令を命令1-1、2-1、3-1と

してその内部に有した非同期イベントタスクの一例であり、図41は、図39の非同期イベントタスクがどのように実行されるかを示すタイミングチャートである。sleep命令が図41に示すように命令1-1として非同期イベントタスクの第1番目の命令として格納されている場合、この命令1-1がDECPC保持部23に格納され、命令解読制御部11により解読されると、セクタ25は、①の信号線を介して命令1-2の読出先アドレスをタスク別PC格納部24に格納する。その後命令解読制御部11は、命令1-2、命令1-3を無効化する(図16における『NOP』)。

【0153】図40はsleep命令及びwake命令の命令フォーマットの一例を示す。図40において、本命令は図9に示したレジスタ指定型の算術演算命令の命令フォーマットを有する。本フォーマットの11bitから13bitを『010』に指定することにより、sleep命令のオペレーションを1/0プロセッサは実行する。また本フォーマットの11bitから13bitを『011』に指定することにより、wake命令のオペレーションを1/0プロセッサは実行する。

【0154】以上のように構成された第5実施形態の1/0プロセッサによって、どのようにスレッドが割り当てられるかを図42(a)～図42(d)を参照しながら説明する。タスク(1)はパーキングタスク、タスク(3)はホスト1/0タスクであり、タスク(1)、タスク(2)、タスク(3)の第1番目の命令がsleep命令であるものとする。タスク(0)、タスク(4)、タスク(5)に4サイクルのスレッドが割り当てられるが、図42(a)においてタスク(1)、タスク(2)、タスク(3)には、第1番目の命令としてsleep命令が格納されているので2サイクルのスレッドが割り当てられる。

【0155】sleep命令の実行後、次のフレームにおいては図42(b)に示すように、タスク(1)、タスク(2)、タスク(3)に割り当てられるスレッドは0サイクルである。タスク(0)は起動期間がより長いビットストリーム転送制御タスクであり、タスク(0)には第1番目にタスク(3)に対してのwake命令が格納されているものとする。そうすると、図42(c)に示すように次のフレームにおいてこのwake命令が解読されることによりタスク(3)に4サイクルのスレッドが割り当てられる。

【0156】タスク(0)には第14番目にタスク(1)に対してのwake命令が格納されているものとする。そうすると、図42(d)に示すように次のフレームにおいてこのwake命令が解読されることによりタスク(1)に4サイクルのスレッドが割り当てられる。以上のように本実施形態によれば、パーキングタスク、ホスト1/0タスク等、起動すべき期間が限定されているタスクは自ら進んで休眠状態に入り、起動期間がより長い転送制御タスクに休眠状態の解除を委ねることにより、タスク巡回のスルーットを向上させることができる。

(第6実施形態)第6実施形態は、第3実施形態にお



るwait状態制御と、第4実施形態における緊急タスク制御と、第5実施形態における休眠タスク制御とを共存させる技術に係る。

【0157】図43は、タスク管理レジスタ13の各ビットにwait状態、emgcy状態、sleep状態の管理を統括させたものである。スレッドマネージャ61及びスケジューラ62は、このようにタスク管理レジスタ13に統括されたwait状態、emgcy状態、sleep状態を参照して次のタスクの決定やタスクの次スレッドに割り当てる命令数の増減を行う。

【0158】図44は、緊急タスク制御及びsleep状態タスク制御を共存させ得よう構成されたスケジューラ62の構成図である。図44においてスケジューラ62がタスクID保持部72、タスクラウンド管理部75、コンバータ76、緊急タスクマスク部77、プライオリティエンコーダ80、セクタ82、タスクID保持部83から構成されている点は第4、第5実施形態と差違は無い。本図において、タスクラウンド管理部75の出力と、インバータ94により反転されたラウンド値taskneのラウンド値との論理和をとるOR回路96が新たに追加され、その結果をプライオリティエンコーダ80に出力する。

【0159】プライオリティエンコーダ80は、OR回路96が出力したラウンド値をタスク識別子に変換して、タスクID保持部72及びタスクID保持部83に出力する。以上のように本実施形態によれば、第4実施形態における緊急タスク制御と、第5実施形態におけるsleep状態タスク制御とを共存させることにより、より柔軟性に富んだタスクスケジューリングを実現することができる。

(各実施形態における全体制御についての説明)以降、第1実施形態～第5実施形態に示したI/Oプロセッサの全体制御をフローチャートを参照しながら説明する。

【0160】図45は、第1実施形態におけるI/Oプロセッサの全体制御を図示したフローチャートである。本図において、『変数k』とは命令メモリ100に記憶されている6本の非同期イベントタスクを指示するための変数でありタスクID保持部71及びタスクID保持部72が保持するタスク識別子taskidに対応する。『変数i』とはそれぞれのタスクに含まれる命令の実行の際、その命令が当該スレッドにおいて何番目に実行されたかを指示するための変数であり、カウンタ52によりカウントされるカウント値に対応する。『変数adr』とは、タスク別PC格納部24に格納されている個々のタスクの読み出し先アドレスを示す変数である。

【0161】図50は、図45のフローチャートによるスレッド間の遷移を示す図である。図50において、縦方向下向きが時間軸であり、4つの長方形を含む太枠で囲まれた四角形がスレッドである。図45のフローチャートにおける変数kにより個別に指示される。太枠の中

の一つの各四角形は一つの命令を実行するサイクルを表し、これらの命令は変数iにより指示される。ステップS1は、上記タスクID保持部71がゼロのタスク識別子taskidを出力する。これにより変数kがゼロクリアされ、ステップS2に移行する。ステップS2は、カウンタ52を初期化することにより変数iを1に初期化する。ステップS3においてタスクID保持部72がタスク別PC格納部24に読み出しアドレス選択信号nexttaskid(rd\_adr)を出力することによりタスク(k)の読出先アドレスをタスク別PC格納部24から読み出す。ステップS4においてタスク(k)の読出先アドレスを用いて、命令メモリ100から命令を取り出す。ステップS5において命令解読制御部11は、取り出された命令を解読して実行する。

【0162】ステップS6は、比較器54は変数iと上限4との比較を行う。この場合はi=1であるのでステップS7に移行する。ステップS7は、カウンタ52に変数iのインクリメントを行わせ、increment回路21に読出先アドレスのインクリメントを行わせるステップであるが、このインクリメントはタスクkにおけるある命令の実行から別の命令の実行への切り換えを意味する。この場合i=2であるから、スレッドにおける命令の実行から第2命令の実行への切り換えが行われることになる。

【0163】変数iは、ステップS7においてインクリメントされてステップS4に移行する。変数iのインクリメントよりi=2となり、読出先アドレスも一命令分進行したので、ステップS4において命令読出回路10は命令メモリ100内の読出先アドレス(adr+1)で指示される命令を読み出して、ステップS5において命令解読制御部11はこれを解読し、実行する。

【0164】以上のステップS4～ステップS7の一連の手順を繰り返すことにより変数iは次々とインクリメントされ、命令メモリ100におけるadr+0, adr+1, adr+2, adr+3の領域に記憶されている命令を順次読み出す。以上のステップS4、ステップS5の実行をステップS6においてYesとなるまで繰り返す。ステップS6がYesとなると、ステップS6からステップS8への移行が行われる。以上の手順において、変数iが1,2,3である内はステップS4、ステップS5、ステップS6、ステップS7の手順が繰り返されて、変数iが4になって初めてステップS6からステップS8への移行が行われたことは、タスク(0)に対する命令実行を4回繰り返すことを意味する(図50において最上段に位置する4つの四角形の並びはタスク(0)の第1スレッドにおける4回の命令実行の図解となる。)。ちなみにステップS8～ステップS10の手順はタスクからタスクへの切り換えを行うものであるから、ステップS6における『変数i=4』の判定は、命令の実行の4回の繰り返しのタスク切り換えの条件に課していることを意味する。

【0165】ステップS6において変数iと上限値との

10

20

30

40

50



判定がなされると、ステップS8においてincrement回路21により変数iがインクリメントされ、ステップS9においてタスク(k)の読出先アドレス(adr+i-1)を読出先アドレス(adr)としてタスク別PC格納部24に格納した後にステップS10に移行する。ステップS10では、タスクラウンド管理部73及びプライオリティエンコーダ74に次に実行すべきタスクを決定させる。この際タスクラウンド管理部73により『000000』のラウンド値tasknが出力され、プライオリティエンコーダ74により、『001』のタスク識別子taskidが出力される。『001』のタスク識別子taskid、即ちk=1となるので、タスク(0)からタスク(1)への切り換えが行われる。タスクの切り換え後、ステップS2に移行すると、変数iの初期化がなされ再度ステップS3～ステップS7の繰り返しが始まる。

【0166】前回同様ステップS4において命令メモリ100における当該読出先アドレスからタスク(1)に含まれる一命令を取り出し、ステップS5において取り出された命令を解釈して実行する。ステップS6は、変数iと上限4との比較を再度行う。この場合はi=1であるのでステップS7に移行する。変数iは、ステップS7においてインクリメントされてステップS4に移行する。変数iのインクリメントよりi=2となるので、ステップS4において読出先アドレス(adr+1)で指示される命令メモリ100内の領域のアクセスを行い、命令メモリ100において、読出先アドレス(adr+1)で指示される領域に記憶されている命令を読み出して、ステップS5においてこれを解釈し、実行する。

【0167】以上のステップS4～ステップS7の一連の手順を繰り返し行うことにより変数iを次々とインクリメントして、命令メモリ100におけるadr+0, adr+1, adr+2, adr+3の領域に記憶されている命令を順次読み出す。以上のステップS4、ステップS5の実行をステップS6においてYesとなるまで繰り返す。ステップS6がYesとなると、ステップS6からステップS8への移行が行われる。以上の手順において、変数iが1,2,3である内はステップS4、ステップS5、ステップS6、ステップS7の手順が繰り返されて、変数iが4になって初めてステップS6からステップS8への移行が行われる。

(以上までの動作により、図50ではタスク(1)における四命令分(タスク(1)に対する4個の命令)進行したことになる。)ステップS8において変数iをインクリメントして、ステップS9においてタスク(k)の読出先アドレス(adr+i-1)を読出先アドレス(adr)として格納した後にステップS10に移行する。ステップS10において再度タスクラウンド管理部73及びプライオリティエンコーダ74に変数kの決定行わせて、変数kをk=2にして、タスク(1)からタスク(2)への切り換えを行わせる。タスクの切り換え後、ステップS2に移行すると、変数iの初期化がなされ再度ステップS3～ステップS

7の繰り返しが始まる。

【0168】以上の繰り返しにより、図50において、タスク(1)、タスク(2)、タスク(3)、タスク(4)、タスク(5)の実行が四命令ずつ行われ、各タスクが順次消費されてゆく。

(第2実施形態における全体制御の説明)第2実施形態におけるI/Oプロセッサの全体制御は、図46のフローチャートに示す手順にて行われる。図46のフローチャートは、ステップS1～ステップS10からなる図45のフローチャートをベースにして作成されている。ここで図46において新規なのはステップS12がステップS4～ステップS5の間に挿入されている点である。

【0169】ステップS12は、ステップS4により命令メモリ100から読み出された命令がWait\_Until\_Next命令であるかを判定する。もしWait\_Until\_Next命令であれば、ステップS8へと移行して変数iをインクリメントし、ステップS9においてタスク(k)の読出先アドレス(adr+i-1)を読出先アドレス(adr)として格納する。

【0170】ここで留意すべきは、ステップS12では変数iの値に拘らず、Wait\_Until\_Next命令が実行された時点でステップS8に移行して読出先アドレスを格納する点である。このようにステップS8が変数iの値に拘らずステップS12の解釈後に実行されるため、タスクからタスクへの切り換えは、四回の命令を待たずに行われることになる。

【0171】図51は、第2実施形態におけるスレッド間の遷移を示す図である。図51におけるシーケンスは、図50同様縦方向下向きが時間軸であり、4つの長方形を含む太枠で囲まれた四角形がスレッドとなる。太枠の中の一つの各四角形は一つの命令を実行するサイクルを表している。これらの四角形のうち、『』が記入されているものはWait\_Until\_Next命令を示す。またタスク(0)が各スレッドにおいて命令メモリ100内のどの命令を実行するかを、四角形の右横の『PC』『PC+1』『PC+2』『PC+3』といったプログラムカウンタ値及びプログラムカウンタからの相対値により表現している(尚本図は、タスク(0)の第1スレッドの第1命令の読み出し時におけるプログラムカウンタ値を基準にしている。)

【0172】図47においてステップS1～ステップS4の手順が行われることにより、タスク(0)の第1スレッドの第1命令が読み出される。読み出された命令の解釈結果の判定がステップS12において行われるが、タスク(0)の第1スレッドー第1命令はWait\_Until\_Next命令ではないのでステップS5に移行する。ステップS5への移行により、タスク(0)の第1スレッドー第1命令の実行が行われて、1命令分進行したことになる。

【0173】再度ステップS1～ステップS4の手順が行われることにより、タスク(0)の第1スレッドの第2命令が読み出される。読み出された命令の解釈結果の判

10

20

30

40

50

定がステップS12において行われるが、タスク(0)の第1スレッド-第2命令はWait\_Until\_Next命令であり、ステップS8への移行が行われる。ステップS8により変数iは1インクリメントされて『3』となり、ステップS9において読出先アドレス(adr+2)がタスク(0)の読み出し先アドレスとしてタスク別PC格納部110に格納される。

【0174】格納後、ステップS10において変数kはインクリメントされ、ステップS2に移行する。このステップS10における変数kのインクリメントにより、タスク(0)からタスク(1)への切り換えが、第1スレッドの第3命令、第4命令の実行を待たずに行われる。タスク(1)の第1スレッドの実行が四命令の実行により行われ、同様にタスク(2)、タスク(3)、タスク(4)、タスク(5)の実行が行われて、タスク(5)の第1スレッド-第4命令まで進んだとする。ステップS10において変数kが決定され、ステップS2において変数iが『1』に初期化されて、ステップS3においてタスク(0)の読出先アドレスがタスク別PC格納部110から読み出される。ここで留意されたいのは、タスク(0)の第1スレッドの第2命令(Wait\_Until\_Next命令)の実行時に、PC+2(『PC』は第1命令のプログラムカウンタ値を示す。)がタスク(0)の読出先アドレスとしてタスク別PC格納部110に格納されている点である。このようにタスク別PC格納部110に(PC+2)が格納されているため、タスク(1)の第2スレッドの命令実行はPC+2から開始されることになる。

【0175】(第3実施形態における全体制御の説明) 図47は第3実施形態におけるI/Oプロセッサのフローチャートである。本図において、図45、図46の同一処理内容のステップには、図45、図46と同一参照符号を付してその説明を省略する。また図中の『変数Total』とは、タスク管理レジスタ13の内容に基づき、セレクタ55が出力する数値を示し、命令実行数の上限を指定するために用いられる。変数Totalはその初期状態において、スレッドにおける命令総数である『4』に初期化されるが、『2』にも更新され得る。

【0176】図52は、第3実施形態のフローチャートによるスレッド間の遷移を示す図である。以降の説明では図52において、タスク(0)の第1スレッドの第1命令、第2命令が実行済みであり、これから第3命令が読み出されようとしている。ステップS93は、上記変数kをゼロクリアし、変数Totalを4に設定する。ステップS2は、上記変数iをゼロクリアする。ステップS3において、ゼロクリアされた変数kの読出先アドレス、即ち、タスク(0)の読出先アドレスをタスク別PC格納部110から読み出す。ステップS4において命令メモリ100において読み出されたタスクの読出先アドレスを絶対アドレスとし、変数iをオフセット値としたアドレスから命令を取り出す。

【0177】ステップS81において取り出された命令

がCmp\_And\_Wait命令であるかを判定する。ステップS82では命令解読制御部81、演算実行部84はCmp\_And\_Wait命令の第1オペランドに記述された状態監視レジスタjの指定と、第2オペランドに記述された即値とを参照してCmp\_And\_Wait命令が提示する事象の成立の真偽を判定する。状態監視レジスタjの保持値と即値とが一致せず、事象不成立が判定されると、ステップS83に移行して、タスクkをwait状態に設定しステップS84に移行する。ステップS84では、プログラムカウンタを進行させないよう、フリップフロップ27及びセレクタ25に読出先アドレスをCmp\_And\_Wait命令のアドレスに設定させてステップS87に移行する。

【0178】ステップS87は、図46のフローチャートのステップS6同様、上限チェックを行うステップである。ステップS6における『変数i=4』の判定は、命令の実行の4回の繰り返しをタスク切り換えの条件に課していた。これに対してステップS87における『変数i=変数Total』の判定は、命令の実行の繰り返し回数が変数Totalと等しいことをタスク切り換えの条件に課している。ここでの変数Totalは未更新であるから、『変数i=変数Total』は実質『変数i=4』の判定となる。Cmp\_And\_Wait命令はタスク(0)の第1スレッドにおいて3命令目に含まれており、一方変数i=2なので、ステップS87からステップS7への移行が行われる。ステップS7において変数iがインクリメントされると、ステップS4における命令読み出しにより、Cmp\_And\_Wait命令の次順の命令が命令解読制御部81に読み出される。読み出されるとステップS81がNoとなりステップS85では、命令解読制御部11は直前の命令において状態不成立が判定され、読出先アドレスがCmp\_And\_Wait命令のアドレスに設定されたか(PC戻りが発生したか)を判定する。

【0179】ここでアドレス(adr+i)に格納されていた命令の直前命令とはCmp\_And\_Wait命令が相当し、このCmp\_And\_Wait命令の解読時にはPC戻りが発生している。そのためステップS85はYesとなりステップS86に移行する。ステップS86では命令解読制御部11は読み出されたアドレス(adr+i)の命令を廃棄するよう、汎用レジスタの格納値を無効化する旨の通知を行い、図51のタスクシーケンスにおけるタスク(0)-第1スレッドの第4命令に示すように、アドレス(adr+i)に格納されていた命令を無効化してステップS87に移行する。変数iの値は4であるからステップS87において『変数i=変数Total』の判定がなされるとYesとなり、ステップS88に移行する。

【0180】ステップS87がYesと判定されたことは、一スレッド分の命令実行が済んだことを意味するが、ステップS88では、その一スレッドにおいてPC戻りが発生したかを判定する。このような判定を行うのはPC戻りが発生した場合はタスク別PC格納部110にその戻

り先を格納しておく必要があるからである。タスク(0)の場合は第4命令が廃棄され、第3命令においてPC戻りが発生しているので、ステップS89においてフリップフロップ27及びセクタ25に戻り先PC値を読出先アドレス(adr)として格納させてステップS10に移行する。ステップS10においてタスクラウンド管理部73及びプライオリティエンコーダ74に次のタスクを決定させた後、変数kをインクリメントしステップS90に移行する。

【0181】ステップS90～ステップS92は、インクリメントされた変数kにより指示されるタスク、即ち次に実行されるべきタスクがwait状態にあるか否かを判定することにより、その次スレッドにおける命令数を切り換えるという重要な意味合いを持つ。順を追って説明するとステップS90では、IDコンバータ53はタスク管理レジスタ13におけるタスク(k)に相当するビットを参照して、タスク(k)がwait状態にあるかを判定する。もしそうであればセクタ55はステップS91において変数Totalに4を設定し、異なれば変数Totalに2を設定する。

【0182】タスク(1)は、wait状態ではないのでステップS91において変数Totalが4に設定されてステップS2に移行する。ここでタスク(1)にはCmp\_And\_Wait命令が含まれていないものとする、ステップS5における命令実行が四回分繰り返されて、次のタスクへの切り換えが行われる(以上の手順は、第1実施形態に記載されたものと同様であるから重複説明はあえて行わない。)

【0183】Cmp\_And\_Wait命令を含んでいないタスク(2)、タスク(3)、タスク(4)についても、四命令の実行が順次行われて順々にタスクが切り換ってゆく。タスク(5)についての四命令の実行が済んでステップS10に移行し、ステップS10において変数kをゼロクリアして、ステップS90に移行したとする。ステップS90では、タスク管理レジスタ13においてタスクkに割り当られたビットkが『1』であるか『0』であるかを判定する。この判定は、タスクkがwait状態であるか否かを見極めを意味する。ここで変数k=0であり、タスク(0)は先程wait状態に設定されているから、ステップS91において変数Totalに『2』を設定する。ここで留意すべきは、変数Totalは、スレッドに割り当てる命令数の意味合いを持っている点である。これをステップS91において『4』から『2』に更新することは、一スレッドにマッピングする命令数を『4』から『2』に削減することを意味する。そのため、タスク(0)の第2スレッドでは、変数iが1,2とインクリメントされている間だけ、つまりステップS81～ステップS85、ステップS5の処理の繰り返しが2回だけとなる。この際、タスク(0)の読出先アドレスとしてタスク別PC格納部110に格納されているのはCmp\_And\_Wait命令の読出先アドレスであ

る。読出先アドレスのアドレスからCmp\_And\_Wait命令が読み出されると第1スレッド同様、ステップS82において事象の成否の判定がなされる。この場合、事象は不成立で終わったものとする、第1スレッドと同様の処理を行って読出先アドレスをCmp\_And\_Wait命令に戻して第2スレッドを終える。

【0184】Cmp\_And\_Wait命令を含んでいないタスク(1)、タスク(2)、タスク(3)、タスク(4)についても、四命令の実行が順次行われて順々にタスクが切り換ってゆき、タスク(5)についての四命令の実行が済んでタスク(0)の第3スレッドが始まろうとしている。ここで1/0プロセッサにおける事象に変化が生じ、タスク(0)のCmp\_And\_Wait命令において指定されていた事象が成立していたとする。このためタスク(0)の第3スレッドでは、ステップS82において命令解読制御部81により事象成立が判定され、ステップS80に移行して、タスク管理レジスタ13におけるタスクkのwait状態が解除される。このようにwait状態が解除されると、ステップS80からステップS87への移行が行われ、ステップS87において変数iと変数Totalとの比較が行われ、ステップS7において変数iのインクリメントがなされて、ステップS4において読出先アドレス(adr+1)から命令が読み出される。ステップS81において読み出された命令がCmp\_And\_Wait命令であるかの判定が行われるが、Cmp\_And\_Wait命令ではないのでステップS85へと移行する。ステップS85では、直前の命令において状態不成立が判定されPC戻りが発生したかが判定される。Cmp\_And\_Wait命令により指定された事象が成立したので、ステップS85がNoとなり、Cmp\_And\_Wait命令の次順命令は二度の事象成立待ちを経てやっとステップS5において実行される。ステップS5における次順命令の実行後、ステップS5からステップS87への移行が行われ、変数iと変数Totalとの大小比較が行われる。但し第2スレッドでは変数Totalが2に設定されているので、タスク(0)の第3スレッドは2回実行されただけで終わってしまう。ステップS87への移行後、ステップS88においては、PC戻りの発生の真偽が判定されるがこの場合はNoであり、ステップS89においてタスク(k)の読出先アドレス(adr+i-1)を読出先アドレス(adr)として格納してステップS10に移行する。このようにしてタスク(0)の第3スレッドが済むと、Cmp\_And\_Wait命令を含んでいないタスク(1)、タスク(2)、タスク(3)、タスク(4)についても、四命令の実行が順次行われて順々にタスクが切り換ってゆく。タスク(5)についての四命令の実行が済んで変数kがゼロクリアされ、ステップS90に移行したとする。

【0185】ステップS90では、タスク管理レジスタ13におけるタスク(k)に相当するビットを参照して、タスク(k)がwait状態にあるかを判定する。この場合タスク(0)のwait状態は解除されたので、ステップS92

において変数Totalに4を設定してステップS2に移行する。このように変数Totalが4に戻ったので、タスク(0)の以降のスレッドは通常通りの四命令ずつ行われることになる。

【0186】(第4実施形態における全体制御の説明) 図48は、第4実施形態におけるI/Oプロセッサの全体制御を示すフローチャートである。図48のフローチャートにおけるステップS1～ステップS9の手順は図46のフローチャートと同様である。図48において新規なのは、30番台の参照符号が付与されたステップS30、ステップS31、ステップS32である。ステップS30は、緊急状態遷移許可信号がLow値であるか否かを判定し、High値ならばステップS31に移行してステップS31において緊急タスクを次に実行すべきタスクとして決定する。

【0187】逆に緊急状態遷移許可信号がLow値ならば、ステップS32に移行してラウンド値taskneに基づいて、通常タスクの中から次に実行すべきタスクをプライオリティエンコーダ80に決定させる。図53は、第4実施形態におけるスレッド間の遷移を示す図である。本図は、通常状態が始まりにおいて、ステップS31の処理がスキップされて、図52のタスク(0)の第1スレッド、タスク(2)の第1スレッド、タスク(3)の第1スレッド、タスク(4)の第1スレッド、タスク(5)の第1スレッドが順次行われる。

【0188】taskneの実行が一巡したところで、緊急状態遷移許可信号のHigh期間が発生したとすると、ステップS2～ステップS9において一つのタスクが実行された後、ステップS30からステップS31への分岐が行われる。分岐後、cstate記憶部85に示されたcstateのHigh値、Low値に応じて緊急タスクの実行を行う。

(第5実施形態における全体制御) 図49は第5実施形態におけるマルチタスク手順のフローチャートである。本図が第1実施形態におけるマルチタスク手順のフローチャートと異なるのは、ステップS4とステップS6との間にステップS41～ステップS44、S46、S47が挿入されている点である。

【0189】ステップS41は、命令メモリ100から読み出された命令がsleep命令であるかを判定する。もしsleep命令であれば、ステップS42へと移行する。ステップS42では、タスク管理レジスタ13はタスク(k)に対応するビットを0に反転する。その後ステップS43において変数iを1インクリメントして、ステップS44においてタスク(k)の読出先アドレス(adr+i-1)を読出先アドレス(adr)として格納した後にステップS45に移行する。移行後、ステップS45では、インバータ94及びOR回路95はタスク管理レジスタ13においてsleep状態が指定されたビット抜きラウンド値を生成し、これをプライオリティエンコーダ74に出力して、次に実行すべきタスク(k)を決定させる。ここで留

意すべきは、先に述べたように『変数k』は命令メモリ100に記憶されている6本のタスクのそれぞれを指示するための変数であり、この決定はタスクからタスクへの切り換えを意味する。

【0190】ここでステップS45がステップS10と異なるのは、ステップS10は変数iが4になった時点で実行されるのに対して、ステップS45は変数iの値に拘らず、sleep命令が解読された時点で実行される点である。このようにステップS45が変数iの値に拘らずsleep命令の解読後に実行されるため、タスクからタスクへの切り換えは、四回の命令を待たずして行われることになる。

【0191】ステップS46では、読出先アドレス(adr+i)から読み出された命令がタスク管理レジスタ内の何れかのビットに対するwake命令であるかを判定する。もし異なれば、ステップS5において読み出された命令を解読して実行するがもしそうであれば、ステップS47において当該wake命令において指定されたjビットを操作することにより、タスクjのsleep状態を解除してステップS6に移行する。

【0192】図54は、第5実施形態におけるスレッド間の遷移を示す図である。本図においてタスク(1)の第1スレッドにおいて印を付しているのは、タスク(1)の第1スレッドの第2命令がsleep\_task命令であることを示している。sleep\_task命令の実行時には、第2実施形態におけるCmp\_And\_Wait命令の実行時と同様、第2命令の解読においてステップS41～ステップS44が実行されてタスク(1)がsleep状態に設定された後にタスク切り換えがなされ、第3命令及び第4命令は事実上廃棄されることになる。

【0193】第1実施形態に述べたようなタスク間の切り換えが順次行われて、タスク(2)、タスク(3)、タスク(4)、タスク(5)というようにタスクの巡回が行われてステップS45に移行したとする。ステップS45においてプライオリティエンコーダ74は、sleep状態にあるタスクのビットがマスクされたラウンド値tasknにより、次に実行すべきタスクを決定する。ここでタスク管理レジスタ13においてタスク(1)に割り当てられているビットは、タスク(1)の第1スレッドの第2命令においてsleep状態に設定されている。そのためタスク(1)の実行はスキップされ、タスク(2)の第2スレッドが続いて実行される。

【0194】同様のタスク間の切り換えが順次行われて、タスク(2)、タスク(3)、タスク(4)、タスク(5)というようにタスクの巡回が行われてステップS45に移行したとする。この巡回において、タスク(5)の第2スレッドの第3命令がタスク(1)のsleep状態解除を指定したwake\_task命令であり、命令メモリ100から読み出されたものとする。wake\_task命令が命令メモリ100から読み出されると、ステップS46の判定においてYes

10

20

30

40

50

となり、ステップS47に移行する。ステップS47では、タスク管理レジスタ13のタスク(1)に割り当てられたビットをオフにすることにより、タスク(1)のsleep状態の解除を行う。解除後、残りの命令をも同様に実行して、第2スレッド分の処理を終える。

【0195】タスク(5)の終了後、タスクがタスク(0)への切り換えられたものとする。またタスク(0)の第3スレッドが実行されて、ステップS45へと移行したとする。ステップS45では、タスク管理レジスタ13においてタスク(1)に割り当てられているビットがオフであるかの判定がなされる。タスク(5)の第2スレッドにおいて、タスク(1)の割り当てビットがオンに設定されたのは先に述べた通りであるから、ステップS45からステップS2、S3への移行が行われて、タスク(1)の実行が再開される。

【0196】

【発明の効果】以上の本発明によれば、 $n$ 個のタスクを実行対象タスクとしたプロセッサであって、 $n$ 個のタスク識別子を出力し、そのタスクに割り当てられた数の命令が実行されると次のタスク識別子を所定の順序で出力する実行タスク指示手段と、出力されるタスク識別子で特定されるタスク中の実行すべき命令を順次指定する命令指定手段と、指定された命令を実行する実行手段とを備えているので、本プロセッサは割込信号の発行を待つのでは無く、オーディオアウトタスク、ビデオアウトタスクを含めた $n$ 個のタスクを所定の命令数ずつ順次実行する。

【0197】このように順次実行されるので、オーディオアウトタスク、ビデオアウトタスクの処理を完遂させるための動作クロック数の最適な下限値を、所定の周期において実行せねばならない命令の最低消化数と、実行周期と、タスク総数とから一義的に導出することができる。それらの最適な下限値から全タスクを動作させるための動作サイクル数の最適な下限値を一義的に計算することができる。

【0198】動作クロック数の最適な下限値が一義的に計算されるため、動作クロック信号を一段と高いものに定めずとも、オーディオアウトタスク、ビデオアウトタスクのリアルタイム性を保証でき、動作クロックが低速なタイプが多い民生機器に搭載される場合でも、MPEGストリームを好適に復号することができる。ここで実行タスク指示手段は、実行された命令をカウントし、カウント値が実行中タスクに割り当てられた命令数になるとタスク切替信号を発生するタスク切替信号発生器と、タスク切替信号が発される度に次順位のタスク識別子を生成して出力するタスク識別子出力部とにより構成することができる。

【0199】ここでタスク識別子出力部は各タスク識別子を何番目に出力するかを示す出力順位を、 $n$ 個の全てのタスクについて記憶する順位記憶部と、タスク切替信

号が発される度に、順位記憶部において出力順位が次順位となるタスク識別子を命令指定手段に出力する選択出力部とにより構成することができる。ここでタスク識別子出力部は、各タスク識別子を何番目に出力するかを示す出力順位を、 $n$ 個の全てのタスクについて記憶する順位記憶部と、タスク切替信号が発される度に、順位記憶部において出力順位が次順位となるタスク識別子を命令指定手段に出力する選択出力部とにより構成することができる。

【0200】ここで何れかのタスクは緊急扱いを要する旨の緊急宣言命令を含み、前記実行タスク指示手段は、プロセッサ外部から入力される所定信号を監視することにより、当該プロセッサ外部が緊急期間内であるか、緊急期間外であるかを判定する監視手段と、所定信号が緊急期間内である場合に、緊急扱いすべきタスクである緊急タスクの識別子を保持する緊急タスクレジスタと、緊急期間外においてタスク切替信号が発されれば、選択出力部が未出力のタスクから選んで出力したタスク識別子を実行手段にスルー出力し、緊急期間内においてタスク切替信号が発されれば、タスク切替信号の $m$ 回の発信につき( $m \geq 2$ )一回の比率にて、緊急タスクレジスタが保持しているタスク識別子を実行手段に出力する出力比率制御部とを構成することができる。

【0201】本構成によれば映像出力に関する非同期イベントタスクの識別子を緊急タスクレジスタに保持させておき、出力比率制御部が緊急期間内にタスク切替信号の $m$ 回の発信につき(一回の比率にて、緊急タスクレジスタが保持しているタスク識別子を出力すれば、ディスプレイにおける水平ブランキング期間、垂直ブランキング期間内にビットストリームを実行させておくことができる。これにより映像信号への伸長を表示期間に間に合わせることができる。このように再生系ハードウェアの処理が周期的にオフになる期間に同期して、効率的に動画データを処理すれば動画データタスクのリアルタイム性を向上させることができる。

【0202】ここで何れかのタスクは自身の実行を休眠状態に設定すべき旨の自発休眠命令を含み、前記実行手段は更に、命令指定手段が次に実行すべき命令として指定した命令を解読する解読手段を備え、前記実行タスク指示手段は更に、解読手段による解読結果が自発休眠命令であれば、当該命令を含むタスクを休眠扱いを要するタスクとしてそのタスク識別子を保持する休眠タスクレジスタを備え、前記選択出力部は、順位記憶部が記憶における次順位のタスク識別子が、休眠タスクのタスク識別子なら、その次順位のタスク識別子を出力するように構成することができる。

【0203】MPEGストリームの復号処理においては、処理が必要となる周期が大きく異なるタスクを並列実行せねばならない。例えば、マクロブロックに含まれている輝度ブロック、色差ブロックをバッファ間で転送させる

ための転送制御用のタスクはマクロブロックにおいて、少なくとも6回以上の頻度にて起動せねばならない。これに対して、MPEGストリームからエレメンタリストリームを抽出するタスクは少ない頻度で起動すればよい。

【0204】休眠タスクレジスタは、タスクに自発休眠命令が含まれていれば、当該命令を含むタスクを休眠扱いを要するタスクとしてそのタスク識別子を保持し、前記選択出力部は、順位記憶部が記憶における次順位のタスク識別子が、休眠タスクのタスク識別子なら、その次順位のタスク識別子を出力するので、実行頻度が少ないタスク実行を省くことができる。これにより非同期イベント処理の起動頻度の差違を考慮して、各非同期イベントタスクに自ら休眠状態に入る機会を与えて、タスクの巡回効率を向上させることができる。

【0205】また、 $n$ 個のタスクを実行対象タスクとしたプロセッサであって、決められたサイクル数でタスクを順次選択するタスク選択手段と、タスクと一対一の関係で $n$ 個の命令指定情報を有し、タスク選択手段によってタスクが選択されるとそれに対応した命令指定情報を有効にすると共に、その情報を起点として、次に読み出すべき命令を指定する情報を動的に生成する命令指定手段と、命令指定手段にて指定された命令を読み出し実行する実行手段と構成することができる。

【0206】ここで命令指定手段は、 $n$ 個のタスクと一対一に対応し、対応するタスクに関して次に読み出すべきアドレス値を命令指定情報として保持している $n$ 個とのアドレスレジスタと、タスク選択手段が選択したタスクに対応するアドレスレジスタを選択し、そのアドレス値を出力させるレジスタ選択部と、レジスタ選択部によりアドレスレジスタが選択されると、当該アドレスレジスタが保持しているアドレス値をカウント初期値として保持するカウント値レジスタと、カウント値レジスタが保持しているカウント値を、サイクル毎に、インクリメントするインクリメントと、インクリメントされたカウント値を前記次に読み出すべき命令を指定する情報の更新値として保持する読出先アドレス保持部とにより構成することができる。

【0207】ここで前記命令指定手段は、インクリメントがインクリメントしたカウント値をスルー出力して読出先アドレス保持部に保持させると共に、タスク選択手段がタスクを選択すると、当該タスクに対応するアドレスレジスタのアドレス値を選択出力して読出先アドレス保持部に保持させる第1セレクタと、次順のタスクに切り換わる際に、カウント値レジスタが保持しているカウント値を用いて、切り換え直前まで選択部で選択されていたアドレスレジスタが格納するアドレス値を書き換える第1書換部とにより構成することができる。

【0208】前記タスク切り換えを指定する旨の自発切換命令が含まれており、前記実行手段は読出先アドレス保持部が保持しているアドレスにて特定される命令を解

読する解読手段を備え、命令指定手段は更に解読手段による解読結果が自発切換命令であれば、読出先アドレス保持部が保持している命令指定情報を用いて、切り換え直前まで選択部で選択されていたアドレスレジスタの格納するアドレス値を書き換える第2書換部を備え、前記第1セレクタは、解読手段による解読結果が自発切換命令であれば、当該次のタスクに対応するアドレスレジスタのアドレス値を選択出力して読出先アドレス保持部に保持させるように構成することができる。

【0209】非同期イベント処理には、内部バッファとSDRAMとの間の入出力を制御するものがあるが、SDRAMの状態は外的要因によって変動することが多い。そのためメモリバリエーションや、あるメモリから値を読み出して、この値を用いた演算を行い、その結果を同一メモリに書き戻すという同一メモリをアクセス先としたメモリアクセス動作は、一つのスレッドにおいて行うのが望ましい。逆に、メモリから値を読み出す読出命令と、この値を用いた演算の演算結果を同一メモリに書き戻す書込命令とが別々のスレッドに配置されてしまうと、1つ目のスレッドが巡ってきってから、2つ目のスレッドが巡ってくる迄の間にメモリ状態が変動してしまい、読出命令は完遂し得たものの、書込命令が完遂できない場合が生じる。

【0210】しかし上記の構成では、自発切換命令を読出命令、書込命令の直前に配置しておけば、第2切り換部がアドレスレジスタの内容を自発切換命令が含まれていたアドレスの次のアドレスに書き換えると共に、第1セレクタが当該次のタスクに対応するアドレスレジスタのアドレス値を選択出力して読出先アドレス保持部に保持させるので、自発切換命令の直後で本タスクの命令実行を打ち切ることができる。即ち、同じタスクのタスク識別子が出力された際、当該タスクは自発切換命令の直後に配置された読出命令、書込命令を、同一のスレッド内で実行することができる。

【0211】これにより、タスク切り換えによる中断を望まない四命令を1タスクに割り当てる等、どの所定数の命令を1スレッドに割り当てるかを自発切換命令の配置により、独自に決めておくことができる。また一部タスクの消化を四命令の実行を待たずに打ち切ることで、他のタスクを早めに消化することもできる。

#### 【図面の簡単な説明】

【図1】AVデコーダの内部構成を示す図である。

【図2】MPEGストリームの階層構造と、AVデコーダの構成要素間の動作タイミングとを示すタイミングチャートである。

【図3】第1実施形態におけるI/Oプロセッサの内部構成を示す図である。

【図4】第1実施形態における命令読出回路10の内部構成を示す図である。

【図5】命令読出回路10内のセレクタ25の出力論理

表を示す図である。

【図6】第1実施形態におけるタスク管理部15の内部構成を示す図である。

【図7】(a)タスク識別子taskidのビット構成を示す図である。

(b)ラウンド値のビット構成を示す図である。

【図8】第1実施形態において命令メモリ100に記憶されている非同期イベントタスクがどのように構成されているかを示す図である。

【図9】レジスタをオペランドに指定した算術演算命令の命令フォーマットを示す。

【図10】レジスタにて読出先アドレス、書込先アドレスを指定したメモリロード命令/メモリストア命令の命令フォーマットを示す。

【図11】8bit、11bitの即値の指定が可能な比較命令/演算命令/分岐命令の命令フォーマットを示す。

【図12】第1実施形態において、IPC、IF1、IF2、DECPCがどのように更新されるかを示すタイミングチャートである。

【図13】タスク(0)からタスク(5)までにどのようにスレッドを割り当てるかを示す図である。

【図14】第2実施形態において命令メモリ100に記憶されている非同期イベントタスクがどのような命令にて構成されているかを示す図である。

【図15】wait\_until\_next命令の命令フォーマットの一例を示す。

【図16】第2実施形態において、IPC、IF1、IF2、DECPCがどのように更新されるかを示すタイミングチャートである。

【図17】第2実施形態においてタスク(0)からタスク(5)までにどのようにスレッドを割り当てるかを示す図である。

【図18】第3実施形態におけるAVデコーダの内部構成を示す図である。

【図19】第3実施形態における命令読出回路10の内部構成を示す図である。

【図20】第3実施形態における命令読出回路10内のセクタ25の出力論理表を示す図である。

【図21】第3実施形態におけるスレッドマネージャ61の内部構成を示す図である。

【図22】タスク管理レジスタ13においてwait状態管理用に割り当られた6bitのビット構成を示す図である。

【図23】第3実施形態における状態監視レジスタのビット構成を示す図である。

【図24】(a)第3実施形態においてどのタスクにCmp\_and\_Wait命令が含まれているかを示す図である。

(b)事象j0,j3,j4,j5が不成立である状態において、図24(a)に示す各タスクにどのようにスレッドが割り当てられるかを示す図である。

(c)事象j0のみが成立したフレームにおいて、図24 50

(a)に示す各タスクにどのようにスレッドが割り当てられるかを示す図である。

(d)事象j0が成立したフレーム以降において、図24(a)に示す各タスクにどのようにスレッドが割り当てられるかを示す図である。

【図25】Cmp\_and\_Wait命令の命令フォーマットの一例を示す。

【図26】Cmp\_and\_Wait命令解読時においてIPC、IF1、IF2、DECPCがどのように更新され、タスク管理レジスタ13のビット及びカウンタの上限値がどう設定されるかを示すタイミングチャートである。

【図27】Cmp\_and\_Wait命令が待っていた事象の未成立時のフレームにおいて、IPC、IF1、IF2、DECPCがどのように更新され、タスク管理レジスタ13のビット及びカウンタの上限値がどう設定されるかを示すタイミングチャートである。

【図28】Cmp\_and\_Wait命令が待っていた事象の成立時のフレームにおいて、IPC、IF1、IF2、DECPCがどのように更新され、タスク管理レジスタ13のビット及びカウンタの上限値がどう設定されるかを示すタイミングチャートである。

【図29】事象の成立時以降のフレームにおいて、IPC、IF1、IF2、DECPCがどのように更新され、タスク管理レジスタ13のビット及びカウンタの上限値がどう設定されるかを示すタイミングチャートである。

【図30】第3実施形態におけるタスク管理部15の内部構成を示す図である。

【図31】第3実施形態におけるスケジューラ62の内部構成を示す図である。

【図32】セクタ82が緊急タスクの識別子を出力する条件を示す図である。

【図33】cstate記憶部85が記憶しているモードの状態遷移図である。

【図34】緊急状態遷移許可信号、タスク切替信号chg\_task\_ex、cstateの遷移に応じて、どのタスク識別子が出力されるかのタイミングを示すタイミングチャートである。

【図35】(a)通常状態において各タスクにどのようにスレッドを割り当てるかを示す図である。

(b)緊急状態において各タスクにどのようにスレッドを割り当てるかを示す図である。

【図36】第4実施形態におけるタスク管理部15の内部構成を示す図である。

【図37】タスク管理レジスタ13においてsleep状態に割り当られたビットの割り当てを示す図である。

【図38】第4実施形態におけるスケジューラ62の内部構成を示す図である。

【図39】第4実施形態において、sleep命令及びwake命令が各タスクにどのように配置されているかを示す図である。

【図40】sleep命令及びwake命令の命令フォーマット  
の一例を示す。

【図41】第4実施形態においてIPC, IF1, IF2, DECPCが  
どのように更新されるかを示すタイミングチャートであ  
る。

【図42】(a)～(d)第4実施形態において、各タ  
スクにどのようにスレッドを割り当てるかを示す図であ  
る。

【図43】第5実施形態におけるタスク管理部15の内  
部構成を示す図である。

【図44】第5実施形態におけるスケジューラ62の内  
部構成を示す図である。

【図45】第1実施形態におけるI/Oプロセッサの全体  
制御を示すフローチャートである。

【図46】第2実施形態におけるI/Oプロセッサの全体  
制御を示すフローチャートである。

【図47】第3実施形態におけるI/Oプロセッサの全体  
制御を示すフローチャートである。

【図48】第4実施形態におけるI/Oプロセッサの全体  
制御を示すフローチャートである。

【図49】第5実施形態におけるI/Oプロセッサの全体  
制御を示すフローチャートである。

【図50】第1実施形態におけるスレッドの割り当てが  
どのように進行するかを示す図である。

【図51】第2実施形態におけるスレッドの割り当てが  
どのように進行するかを示す図である。

【図52】第3実施形態におけるスレッドの割り当てが  
どのように進行するかを示す図である。

【図53】第4実施形態におけるスレッドの割り当てが  
どのように進行するかを示す図である。

【図54】第5実施形態におけるスレッドの割り当てが  
どのように進行するかを示す図である。

【図55】第1実施形態におけるスレッド割り当てにて  
ビデオアウトタスク、オーディオアウトタスクを実行す  
る場合に動画ストリーム、オーディオストリームがど  
のように処理されるかを示すタイミングチャートである。

【図56】タスク別にサイクル数を可変にする場合のス  
レッドマネージャの構成図である。

【図57】割込信号にてオーディオアウトタスク、ビデ  
オアウトタスクを実行する場合に動画ストリーム、オー  
ディオストリームの処理を何時までに完遂せねばなら  
ないかを示すタイミングチャートである。

【符号の説明】

10 命令読出回路

11 命令解読制御部

12 レジスタセット

13 タスク管理レジスタ

14 演算実行部

15 タスク管理部

20 IF1+1保持部

21 increment回路

22 IF2保持部

23 DECPC保持部

24 タスク別PC格納部

25 4入力-1出力セクタ

26 2入力-1出力セクタ

52 カウンタ

10 53 コンバータ

54 比較器

55 セクタ

61 スロットマネージャ

62 スケジューラ

71 タスクID保持部

72 タスクID保持部

73 タスクラウンド管理部

74 プライオリティエンコーダ

75 タスクラウンド管理部

20 76 コンバータ

77 緊急タスクマスク部

80 プライオリティエンコーダ

81 命令解読制御部

82 セクタ

83 タスクID保持部

84 演算実行部

85 CSTATE記憶部

93 タスクスキップ管理回路

94 インバータ

30 100 命令メモリ

101 ストリーム入力部

102 バッファメモリ

104 Setup部

105 VLD部

106 IQ/IDCT部

107 動き補償部

108 ビデオ出力部

109 オーディオ出力部

110 BMバッファメモリコントローラ

40 111 RAMコントローラ

112 FIFOコントローラ

113 プロセッサ

115 ホストI/O部

121 バッファメモリバス

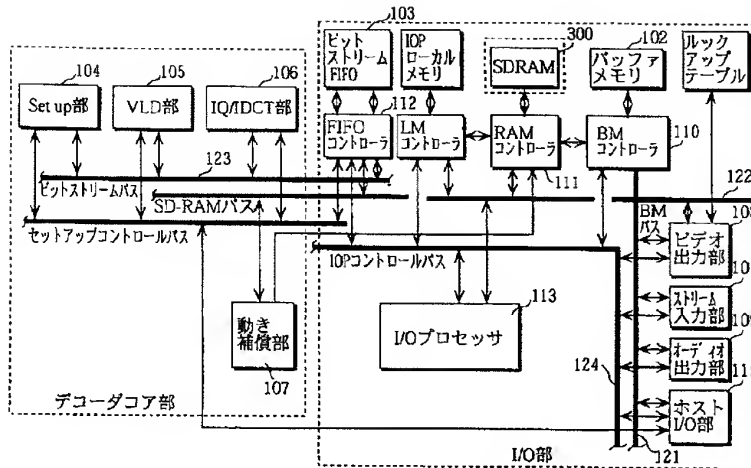
122 SD-RAMバス

123 ビットストリームバス

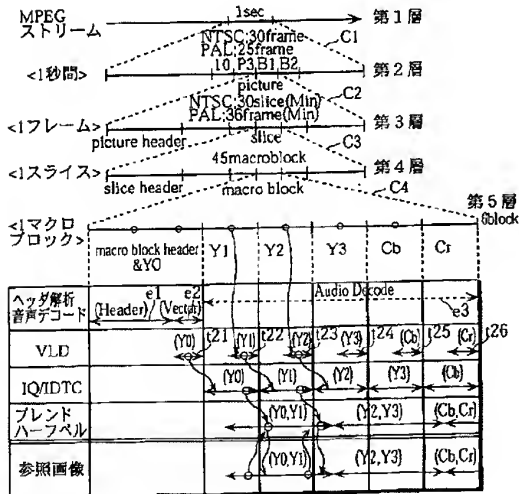
124 コントロールバス



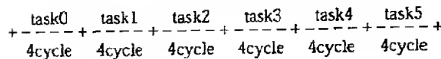
【図1】



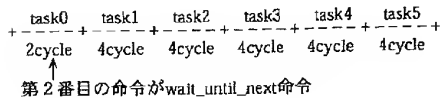
【図2】



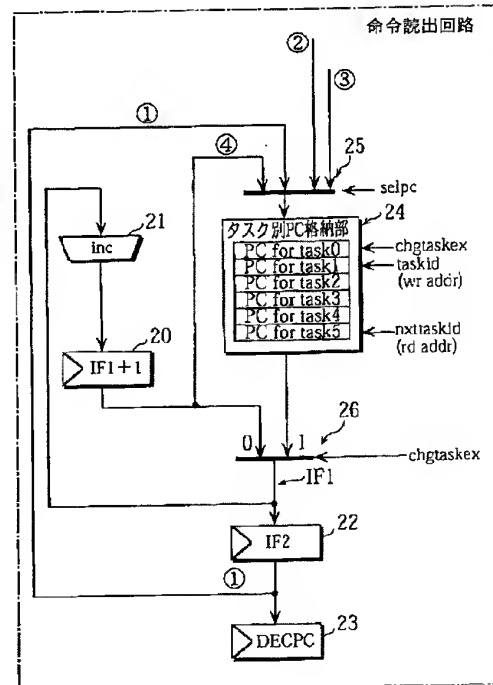
【図13】



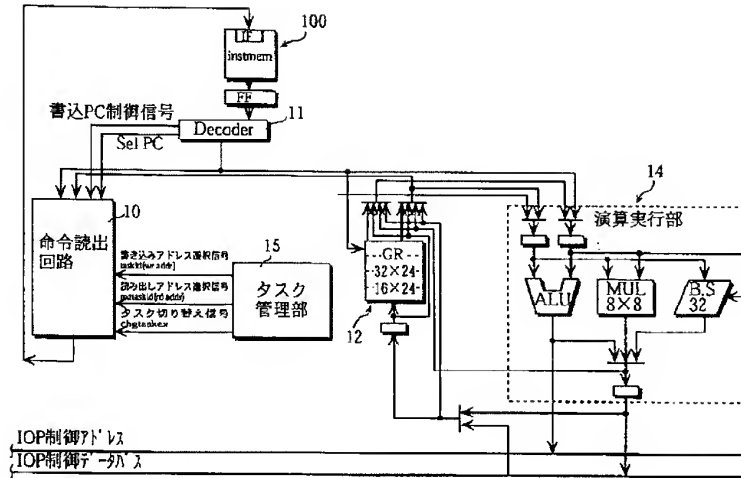
【図17】



【図4】



【図3】

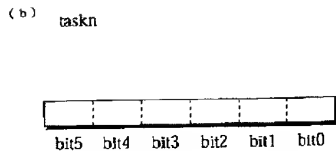
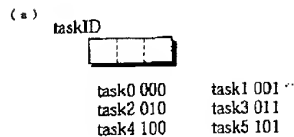


【図5】

セレクタ25の出力論理

wait_until_next命令の解釈時	セレクタ26から出力されたアドレス(ルート①)
絶対アドレス型分岐命令の解釈時	デコーダ11から出力されたアドレス(ルート②)
間接アドレス型分岐命令の解釈時	ALU14から出力されたアドレス(ルート③)
chg_task_id信号の出力時	IF2から出力されたアドレス(ルート④)

【図7】

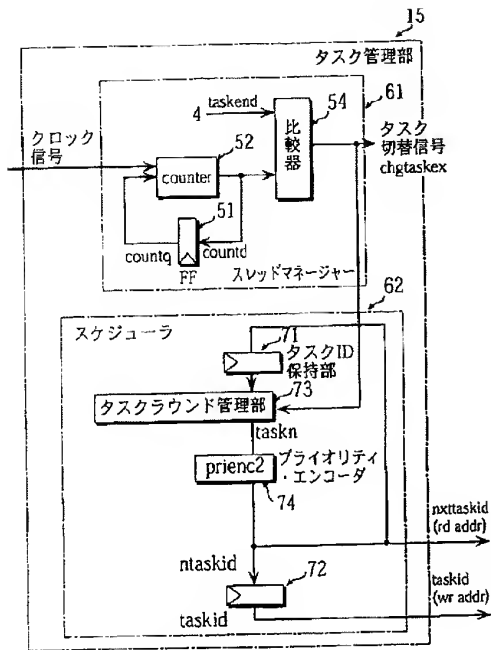


bit0 . . . 1 : task0が実行済み 0 : 未  
 bit1 . . . 1 : task1が実行済み 0 : 未  
 bit2 . . . 1 : task2が実行済み 0 : 未  
 bit3 . . . 1 : task3が実行済み 0 : 未  
 bit4 . . . 1 : task4が実行済み 0 : 未  
 bit5 . . . 1 : task5が実行済み 0 : 未

【図8】

task0	task1	task2	task3	task4	task5
命令0-0	命令1-0	命令2-0	命令3-0	命令4-0	命令5-0
命令0-1	命令1-1	命令2-1	命令3-1	命令4-1	命令5-1
命令0-2	命令1-2	命令2-2	命令3-2	命令4-2	命令5-2
命令0-3	命令1-3	命令2-3	命令3-3	命令4-3	命令5-3
命令0-4	命令1-4	命令2-4	命令3-4	命令4-4	命令5-4
命令0-5	命令1-5	命令2-5	命令3-5	命令4-5	命令5-5
命令0-6	命令1-6	命令2-6	命令3-6	命令4-6	命令5-6
命令0-7	命令1-7	命令2-7	命令3-7	命令4-7	命令5-7
命令0-8	命令1-8	命令2-8	命令3-8	命令4-8	命令5-8
命令0-9	命令1-9	命令2-9	命令3-9	命令4-9	命令5-9
命令0-10	命令1-10	命令2-10	命令3-10	命令4-10	命令5-10
命令0-11	命令1-11	命令2-11	命令3-11	命令4-11	命令5-11
命令0-12	命令1-12	命令2-12	命令3-12	命令4-12	命令5-12
命令0-13	命令1-13	命令2-13	命令3-13	命令4-13	命令5-13
命令0-14	命令1-14	命令2-14	命令3-14	命令4-14	命令5-14
命令0-15	命令1-15	命令2-15	命令3-15	命令4-15	命令5-15

【図 6】



【图 14】

task0	task1	task2	task3	task4	task5
命令0-0	命令1-0	命令2-0	命令3-0	命令4-0	命令5-0
命令0-1 wun 0-1	命令1-1	命令2-1	命令3-1	命令4-1	命令5-1
命令0-2 LD mem1,r0	命令1-2	命令2-2	命令3-2	命令4-2	命令5-2
命令0-3 LD buf1,r1	命令1-3	命令2-3	命令3-3	命令4-3	命令5-3
命令0-4 mul r0,r1,r2	命令1-4	命令2-4	命令3-4	命令4-4	命令5-4
命令0-5 ST r2,mem1	命令1-5	命令2-5	命令3-5	命令4-5	命令5-5
命令0-6	命令1-6	命令2-6	命令3-6	命令4-6	命令5-6
命令0-7	命令1-7	命令2-7	命令3-7	命令4-7	命令5-7
命令0-8	命令1-8	命令2-8	命令3-8	命令4-8	命令5-8
命令0-9	命令1-9	命令2-9	命令3-9	命令4-9	命令5-9

【図9】

## レジスタ指定型算術演算命令

0	0	src1	src2	dst/cond	operation
0	1	2	3	4	5
6	7	8	9	10	11
12	13	14	15		
operation					
high		00		01	
low		00		01	
000		src1+src2=dst ADD		src1-src2=dst SUB	
001		src1<<src2=dst		src1>>src2=dst MUL	
010		src1 and src2=dst AND		src1 or src2=dst OR	
011		src1 andn src2=dst ANDN		src1 orn src2=dst ORN	
100		src1=dst MOV		-src1=dst NOT	
101		Mask G(src1)-dst MASKG		NOP NOP	
110		BIN0		BIN1 BIN2	
111		BEX0		BEX1 BEX2	
				bra src1,pc+1=dst BRA	
				BIN3 BEX3	

【図 25】

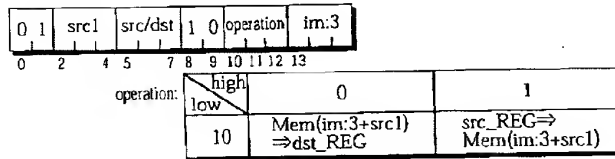
0	1	cond			cr_state	0	imm:5							
2	4	5	7	9	10	11	15							

<div style="display: flex; align-items: center; justify-content: center;"> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">high low</div> <div style="border: 1px solid black; padding: 2px; flex-grow: 1;">0</div> <div style="border: 1px solid black; padding: 2px; flex-grow: 1;">1</div> </div>											
		0	cr_state cmp imm:5					cr_st cmp & wait			

【図10】

メモリロード/メモリストア命令

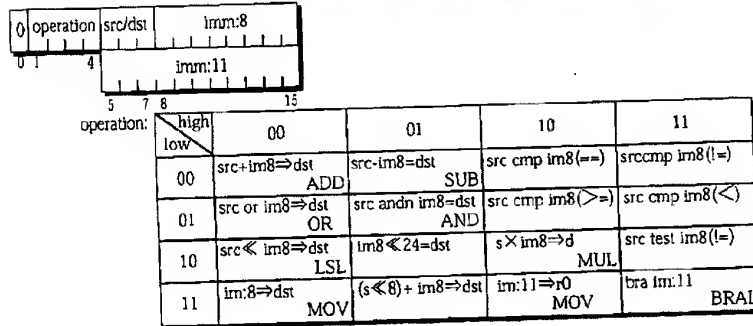


【図32】

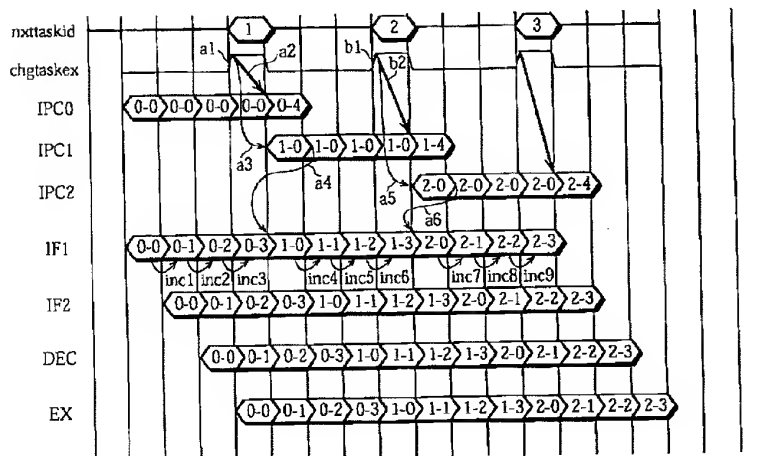
lexecmode	!cstate	セクタ52の出力
1	0	taskide
0	1	taskidne
0	0	taskidne
1	1	taskidne

【図11】

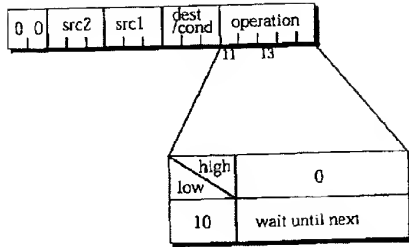
即値付き比較命令/即値付き演算命令/絶対アドレス指定分岐命令



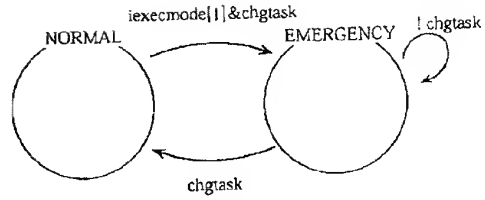
【図12】



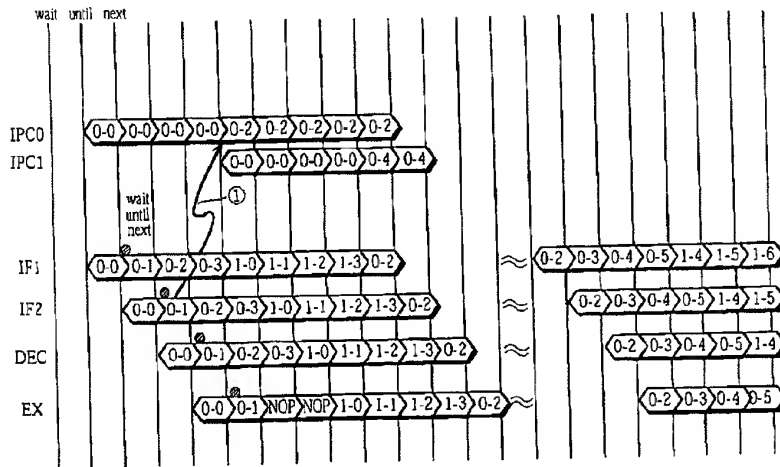
【図15】



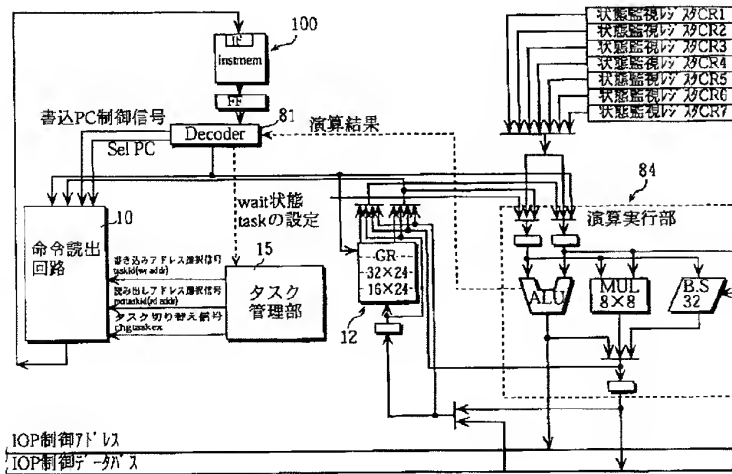
【図13】



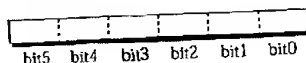
【図16】



【図18】



【图 2-2】



```

bit0 . . . 1 : task0を wait 状態に 設定 0 : 未
bit1 . . . 1 : task1を wait 状態に 設定 0 : 未
bit2 . . . 1 : task2を wait 状態に 設定 0 : 未
bit3 . . . 1 : task3を wait 状態に 設定 0 : 未
bit4 . . . 1 : task4を wait 状態に 設定 0 : 未
bit5 . . . 1 : task5を wait 状態に 設定 0 : 未

```

【图 20】

wait_until_next命令の解読時	セクタ26から出力されたアドレス(ルート①)
絶対アドレス型分岐命令の解読時	デコーダ11から出力されたアドレス(ルート②)
間接アドレス型分岐命令の解読時	ALU14から出力されたアドレス(ルート③)
chg_task_id信号の出力時	IF2から出力されたアドレス(ルート④)
cmp_and_wait命令の解読時	FF27から出力されたアドレス(ルート⑤)

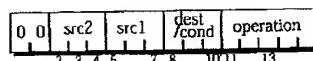
【图 40】

bit5 bit4 bit3 bit2 bit1 bit0

```

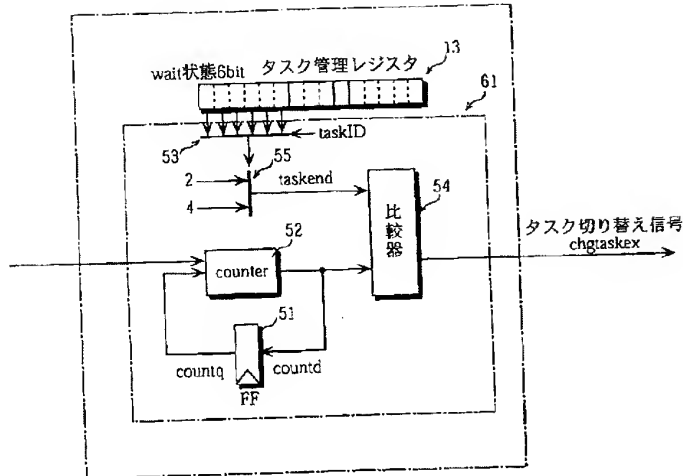
bit0  . . 0 : task0を sleep 状態に設定 1: 未
bit1  . . 0 : task1を sleep 状態に設定 1: 未
bit2  . . 0 : task2を sleep 状態に設定 1: 未
bit3  . . 0 : task3を sleep 状態に設定 1: 未
bit4  . . 0 : task4を sleep 状態に設定 1: 未
bit5  . . 0 : task5を sleep 状態に設定 1: 未

```

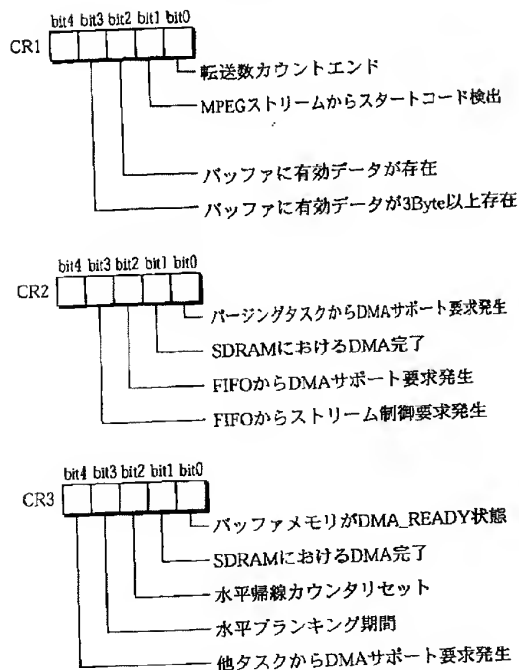


010	Sleep src1/src2
011	wake src1/src2

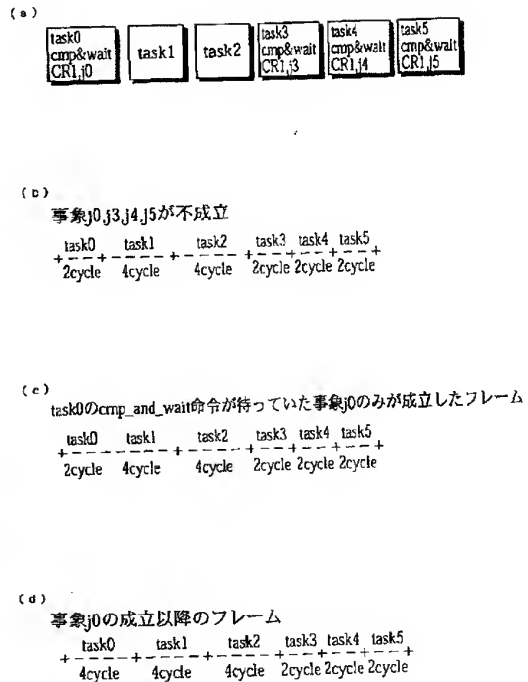
【図21】



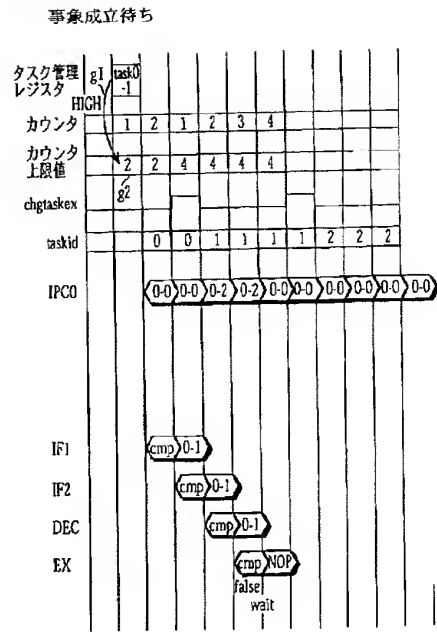
【図23】



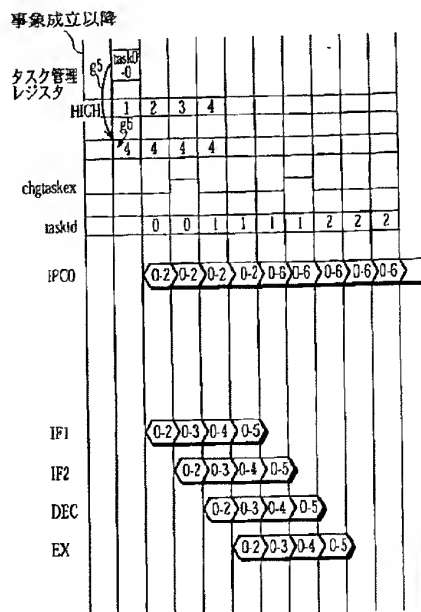
【図24】



【図 27】

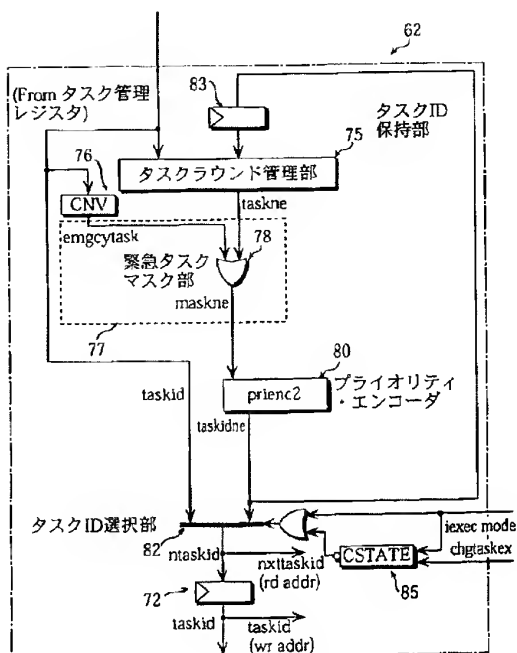


【图 29】

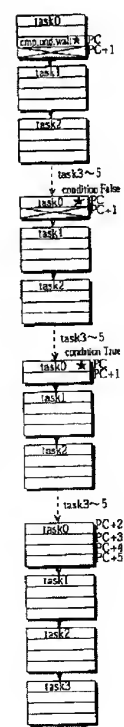




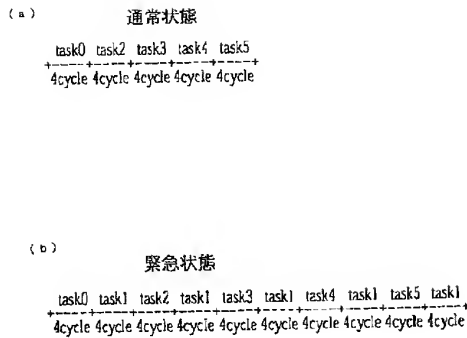
【図 3 1】



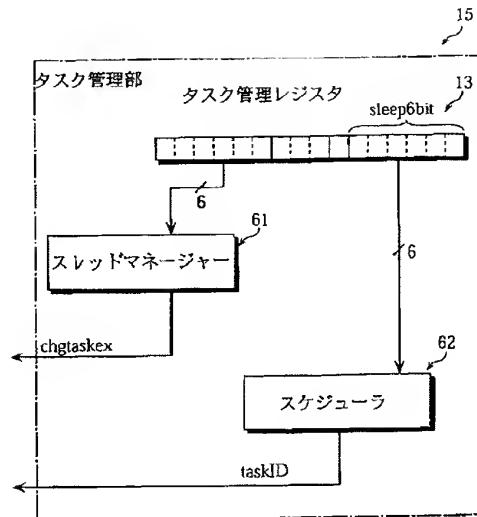
【图 5 2】



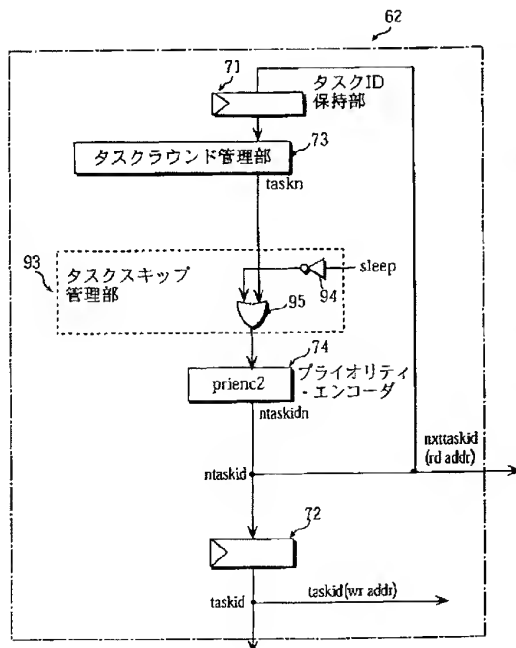
【図35】



【図36】



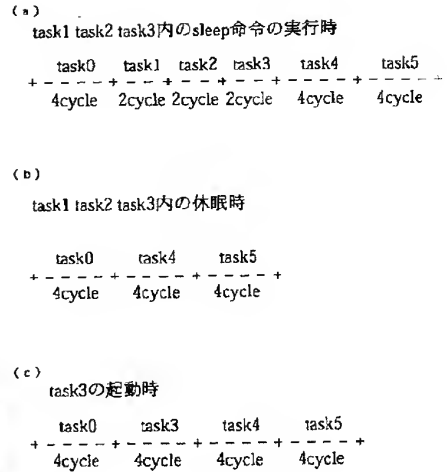
【図38】



【図39】

task0	task1	task2	task3	task4	task5
メモリ制御	パージング		HOST I/F		
命令0-0	命令1-0	命令2-0	命令3-0	命令4-0	命令5-0
命令0-1	命令1-1	命令2-1	命令3-1	命令4-1	命令5-1
命令0-2	命令1-2	命令2-2	命令3-2	命令4-2	命令5-2
命令0-3	命令1-3	命令2-3	命令3-3	命令4-3	命令5-3
命令0-4	命令1-4	命令2-4	命令3-4	命令4-4	命令5-4
命令0-5	命令1-5	命令2-5	命令3-5	命令4-5	命令5-5
命令0-6	命令1-6	命令2-6	命令3-6	命令4-6	命令5-6
命令0-7	命令1-7	命令2-7	命令3-7	命令4-7	命令5-7
命令0-8	命令1-8	命令2-8	命令3-8	命令4-8	命令5-8
命令0-9	命令1-9	命令2-9	命令3-9	命令4-9	命令5-9
命令0-10	命令1-10	命令2-10	命令3-10	命令4-10	命令5-10
命令0-11	命令1-11	命令2-11	命令3-11	命令4-11	命令5-11
命令0-12	命令1-12	命令2-12	命令3-12	命令4-12	命令5-12
命令0-13	命令1-13	命令2-13	命令3-13	命令4-13	命令5-13
命令0-14	命令1-14	命令2-14	命令3-14	命令4-14	命令5-14
命令0-15	命令1-15	命令2-15	命令3-15	命令4-15	命令5-15

【图 4 2】

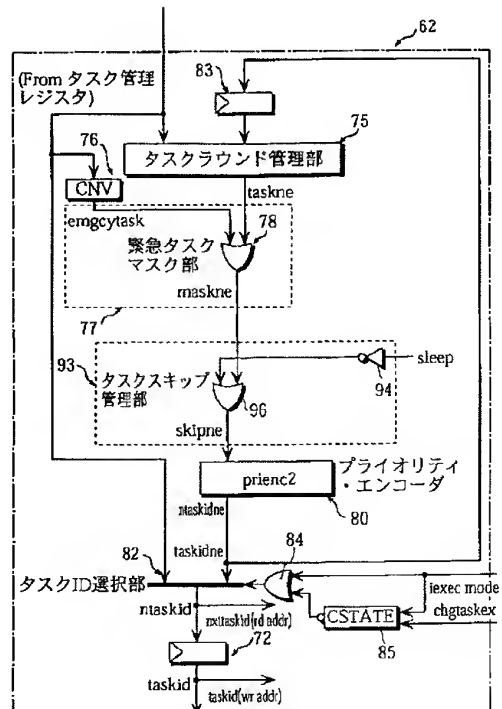


(d)

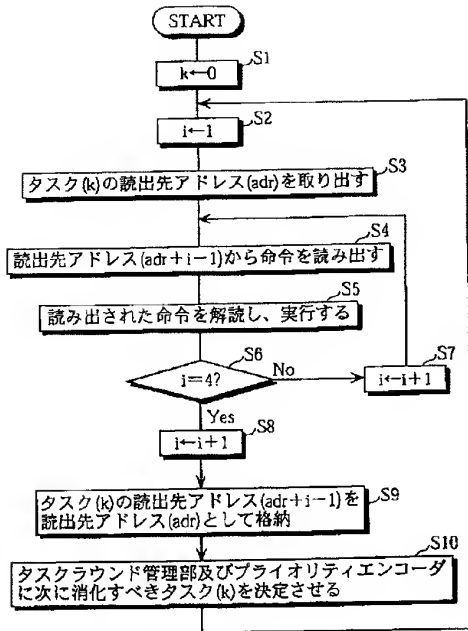
task1の起動時

+ task0 + task1 + task3 + task4 + task5 +

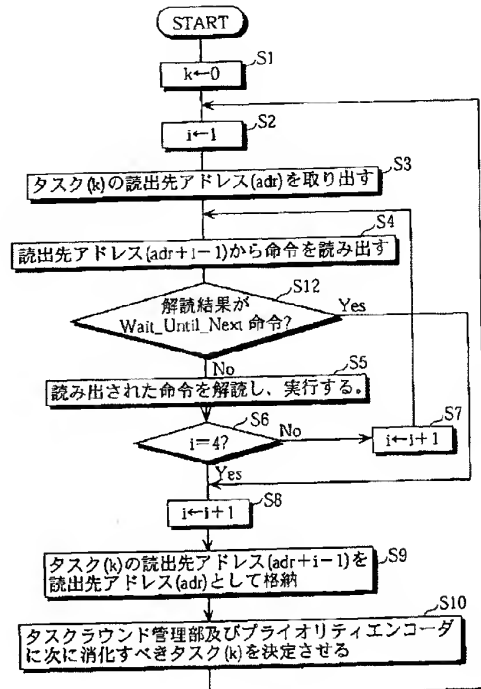
4cycle 4cycle 4cycle 4cycle 4cycle



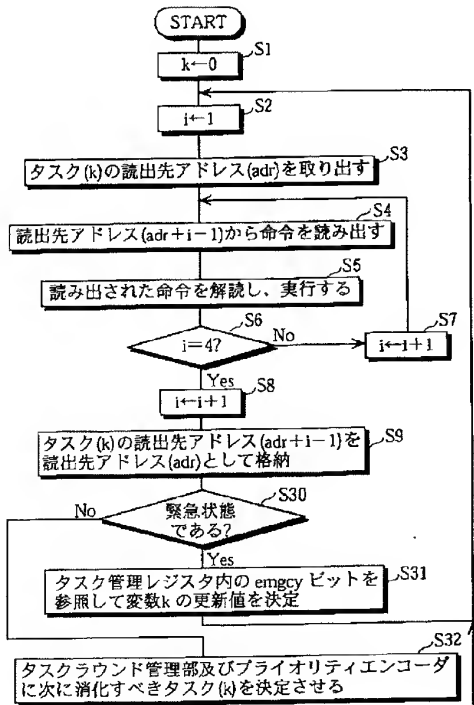
【図45】



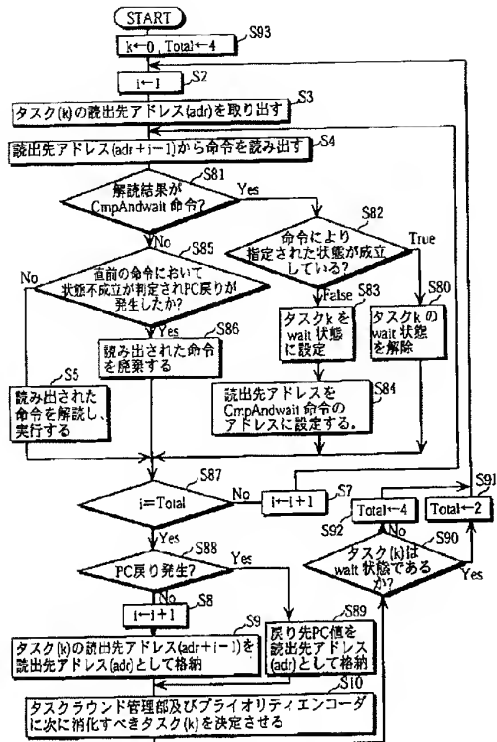
【図46】



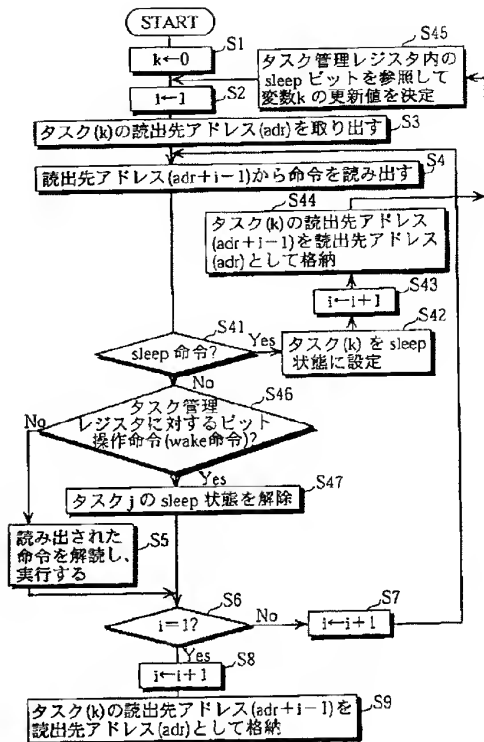
【図48】



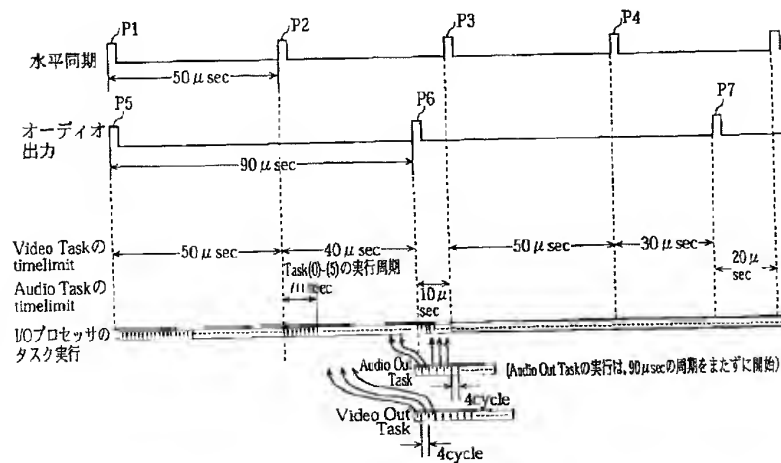
【図47】



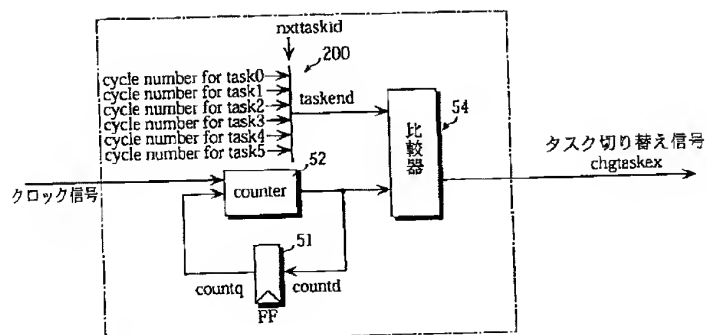
【図49】



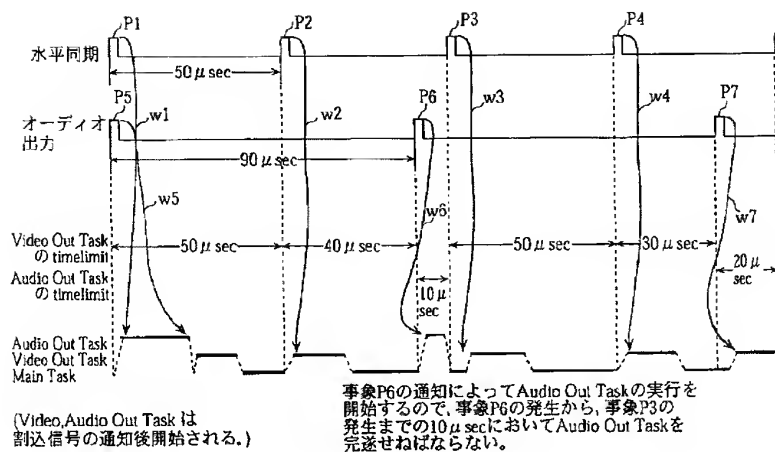
【図55】



【図 5 4】



【図57】



フロントページの続き

(72)発明者 平井 誠  
大阪府門真市大字門真1006番地 松下電器  
産業株式会社内

(72)発明者 清原 督三  
大阪府門真市大字門真1006番地 松下電器  
産業株式会社内